arXiv:2306.17253v1 [cs.CV] 29 Jun 2023

Towards Zero-Shot Scale-Aware Monocular Depth Estimation

Vitor Guizilini Igor Vasiljevic Dian Chen Rareș Ambruș Adrien Gaidon

Toyota Research Institute (TRI), Los Altos, CA

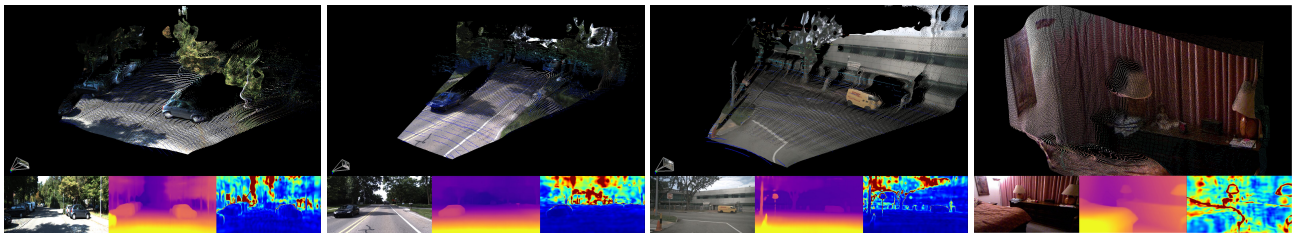


Figure 1: **Our proposed framework enables robust zero-shot transfer of metric depth predictions**. The pointclouds above were generated by *the same model*, that *has never seen any of these datasets*, and *without groundtruth scale alignment*. Ground-truth LiDAR pointclouds are shown as height maps, overlaid with colored predicted monocular pointclouds.

Abstract

Monocular depth estimation is scale-ambiguous, and thus requires scale supervision to produce metric predictions. Even so, the resulting models will be geometry-specific, with learned scales that cannot be directly transferred across domains. Because of that, recent works focus instead on relative depth, eschewing scale in favor of improved up-to-scale zero-shot transfer. In this work we introduce ZeroDepth, a novel monocular depth estimation framework capable of predicting metric scale for arbitrary test images from different domains and camera parameters. This is achieved by (i) the use of input-level geometric embeddings that enable the network to learn a scale prior over objects; and (ii) decoupling the encoder and decoder stages, via a variational latent representation that is conditioned on single frame information. We evaluated ZeroDepth targeting both outdoor (KITTI, DDAD, nuScenes) and indoor (NYUv2) benchmarks, and achieved a new state-of-the-art in both settings using the same pre-trained model, outperforming methods that train on in-domain data and require test-time scaling to produce metric estimates. Project page: https://sites.google.com/view/tri-zerodepth.

1. Introduction

Monocular depth estimation is a key task in computer vision, with practical applications in areas such as robotics [9, 32] and autonomous driving [19, 23, 39, 27]. It is easy to understand why: the promise of turning any camera into a *dense* range sensor is very appealing, both as a means to reduce costs and in terms of its rich semantics and widespread application. However, in order to be truly useful as a 3D reconstruction tool these predictions need to be *scale-aware*, meaning that they need to be metrically scaled. Supervised methods train with groundtruth depth maps [14, 40], while self-supervised methods inject additional information in the form of velocity measurements [23], camera intrinsics [2] and/or extrinsics [27, 60]. Even so, the resulting models will be camera-specific, since the learned scale will not transfer across datasets, due to differences in the cameras used to capture training data.

This *geometric domain gap* is separate from the traditional *appearance domain gap*, however while the latter has been extensively studied in recent years [26, 63, 29, 41, 57, 36, 64], very few works have addressed the former [2, 12, 61]. Instead, the recent trend is to focus on relative depth [10, 50, 51], eschewing scale completely in favor of improved zero-shot transfer of unscaled depth predictions. Even though this approach qualitatively leads to very accurate depth maps, the resulting predictions still require groundtruth information at test-time to be metrically scaled, which severely limits their application in practical scenarios, such as autonomous driving and indoor robotics.

In this paper, we rethink this recent trend and introduce *ZeroDepth*, a novel monocular depth estimation framework that is robust to the geometric domain gap, and thus capable of generating metric predictions across different datasets. We achieve this by proposing two key modifications to the standard architecture for monocular depth estimation: (i) we use input-level geometric embeddings to jointly encode camera parameters and image features, which enables the network to reason over the physical size of objects and learn scale priors; and (ii) we decouple the encoding and decoding stages, via a learned global latent representation. Im-

portantly, this latent representation is *variational*, and once conditioned can be sampled and decoded to generate multiple predictions in a probabilistic fashion. By training on large amounts of scaled, labeled data from real-world and synthetic datasets, our framework learns depth and scale priors anchored in physical 3D properties that can be directly transferred across datasets, resulting in the zero-shot prediction of metrically accurate depth estimates. In summary, our contributions are as follows:

- We introduce **ZeroDepth**, a novel variational monocular depth estimation framework capable of **transferring metrically accurate predictions** across datasets with different camera geometries.
- We propose a series of **encoder-level data augmentation techniques** aimed at improving the robustness of our proposed framework, addressing both the **appearance and geometric domain gaps**.
- As a result, **ZeroDepth achieves state-of-the-art zero-shot transfer** in both outdoor (KITTI, DDAD, nuScenes) and indoor (NYUv2) benchmarks, outperforming methods that require in-domain training images and test-time ground truth scale alignment.

2. Related Work

2.1. Monocular Depth Estimation

Monocular depth estimation is the task of estimating per-pixel distance to the camera based on a single image. Early learning-based approaches were fully supervised [11], requiring datasets collected using additional range sensors such as IR [48] or LiDAR [17]. Although these methods naturally produce metric predictions, they suffer from sparsity and high noise levels in the "groundtruth" training data, as well as limited scalability due to the need of dedicated hardware and calibration. The work of Zhou *et al.* [65] introduced the concept of self-supervised monocular depth estimation, that eliminates explicit supervision in favor of a multi-view photometric objective. Further improvements in this setting have led to accuracy that competes with supervised approaches [19, 20, 23, 25, 22]. However, because the multi-view photometric objective is scale-ambiguous, such models are typically evaluated by aligning predictions to groundtruth depth at test time (typically *median-scaling* [65, 18]), at the expense of practicality.

2.2. Scale-Aware Monocular Depth Estimation

To address the inherent scale-ambiguity in self-supervised monocular depth estimation, several works have looked into ways to inject indirect sources of metric information. In [23], the authors use velocity measurements as weak supervision to jointly train scale-aware depth and pose networks. In [58], the camera height is used at training time in conjunction with a ground plane segmentation network to generate scaled predictions. FSM [27] uses known camera extrinsics with arbitrary overlaps to enforce spatio-temporal photometric constraints at training time, leading to improvements in depth estimation as well as scaled predictions. SurroundDepth [60] also uses known camera extrinsics and proposes a joint network to process surrounding views, as well as a cross-view transformer to effectively fuse multi-view information. Similarly, VolumetricFusion [37] constructs a volumetric feature map by extracting feature maps from surround-view images, and fuse feature maps into an unified 3D voxel space. DistDepth [61] uses left-right stereo consistency to distill structure information and metric scale into an off-the-shelf scale-agnostic depth network, focusing on indoor datasets.

2.3. Zero-Shot Monocular Depth Estimation

The observation that models trained with in-domain data will overfit to the camera geometry, along with limitations in the self-supervised photometric objective [24, 19, 22, 13], have led to a recent emphasis on *zero-shot* depth estimation, in which a pre-trained model is evaluated on out-of-domain data without fine-tuning. To achieve such robustness to domain shifts, these methods rely on large-scale and diverse training data, usually from multiple sources, and propose different ways to encode geometry. In [12], the authors propose CAM-Convs as a way to inject camera parameters into the convolutional operation, resulting in calibration-aware features. An alternative approach is described in [2], which resizes and crops input images to conform to fixed camera parameters, thus abstracting geometry away from the learned features. Another way to abstract away geometry is proposed in [51], that uses scale-invariant losses in combination with heterogeneous datasets to achieve impressive qualitative results, albeit unscaled. This approach is further explored in [10], that proposes a novel pipeline for the generation of additional synthetic training data.

ZoeDepth [4] is a concurrent monocular depth estimation work that also claims the zero-shot transfer of metrically accurate predictions. This is achieved by fine-tuning a scale-invariant model on a combination of indoor and outdoor datasets, and predicting domain-specific adaptive ranges. However, as we show in experiments, this leads to specialization to these training domains, as well as to their camera geometries. ZeroDepth instead directly decodes metric depth without adaptive range prediction, and thus is not bounded or conditioned to any specific domain.

3. Zero-Shot Scale-Aware Monocular Depth

3.1. Perceiver IO Overview

Perceiver IO [33] is an efficient Transformer architecture that alleviates one of the main weaknesses of Transformer-

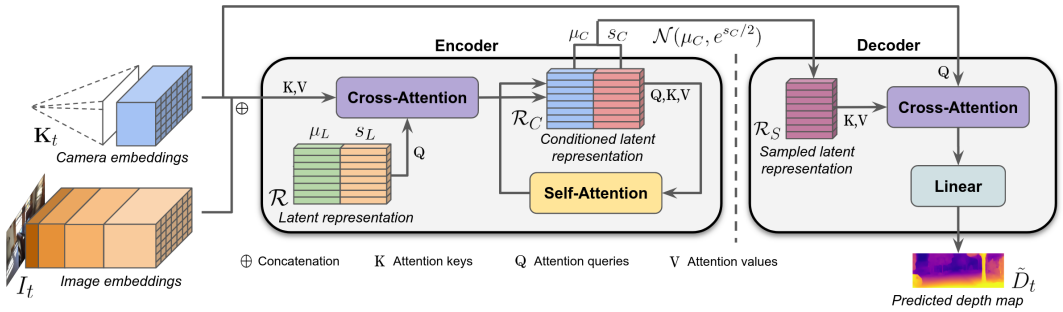


Figure 2: **Diagram of our proposed ZeroDepth framework.** During the encoding stage, an input frame I_t with intrinsics $\mathbf{K}_t$ is processed to generate image $\mathcal{E}_I$ and geometric $\mathcal{E}_G$ embeddings. These are concatenated and used to condition our variational latent representation, that can then be sampled and decoded to generate predictions using only geometric embeddings.

based methods [56], namely the quadratic scaling of self-attention with input size. This is achieved by learning a $N_l \times D_l$ *latent representation* $\mathcal{R}$, and projecting $N_e \times D_e$ encoding embeddings onto this latent representation using cross-attention. Self-attention is performed in this lower-dimensional space, producing a *conditioned latent representation* $\mathcal{R}_c$ that is queried using $N_d \times D_d$ decoding embeddings to generate estimates. This architecture has been successfully applied to multi-frame tasks such as optical flow [33], stereo [62, 28], and video depth estimation [28].

3.2. The ZeroDepth Framework

Our ZeroDepth framework generalizes the Perceiver IO architecture, proposing two key modifications relative to prior works that use it for depth estimation [62, 28]. Firstly, we focus on the *monocular* setting, leveraging input-level inductive biases not to learn implicit multi-view geometry, but rather scale priors that can be transferred across datasets. By augmenting image features with camera information, we enable our model to implicitly reason over physical properties such as size and shape, that are more robust to the geometric domain gap. Secondly, we maintain a *variational latent representation*, that after conditioning results in a distribution which can be sampled during the decoding stage to generate estimates in a probabilistic fashion. Our hypothesis is that, given the extreme diversity in training datasets both in terms of appearance and geometry, the entropy of possible depth predictions is too high to be modelled as a single point estimate, and hence we model it instead as a probability distribution. A diagram of ZeroDepth is shown in Figure 2, and below we describe in details each of its components.

3.3. Input-Level Embeddings

Image Embeddings. We use a ResNet18 [30] backbone as the image encoder, taking as input an $H \times W \times 3$ image I_t and producing a list of feature maps at increasingly lower resolutions and higher dimensionalities. Following [28], feature maps at $1/4$ the original resolution are concatenated with bilinearly upsampled lower-resolution feature maps, resulting in $H/4 \times W/4 \times 960$ image embeddings $\mathcal{E}_I$ that are used to encode frame-specific visual information onto the latent representation $\mathcal{R}$.

Geometric Embeddings. We augment image embeddings with camera information, as a way to generate *geometry-aware features* capable of reasoning over the physical shape of objects. For simplicity, we assume pinhole cameras with 3×3 intrinsics matrix $\mathbf{K}_t$. The viewing direction of a pixel $\mathbf{p}_{ij} = [u_{ij}, v_{ij}]$ is given by $\mathbf{r}_t^{ij} = \mathbf{K}_t^{-1} [u_{ij}, v_{ij}, 1]^T$. This vector is normalized and Fourier-encoded [62] to produce pixel-level $3(F+1)$-dimensional geometric embeddings $\mathcal{E}_G$, where F is the number of frequency bands. Note that camera centers (and, by extension, poses) are not required, since we operate on a single-frame setting, and thus each sample is at the origin of its own coordinate system. During the encoding stage, camera parameters are scaled down to $1/4$ of the original resolution, to match image embeddings. During the decoding stage the original resolution can be used, since only geometric embeddings are required.

3.4. Variational Latent Representation

Variational inference [6] is a powerful statistical tool that provides a tractable way to approximate difficult-to-compute probability densities using optimization. Given input embeddings $\mathcal{E}$, the posterior over our latent representation $\mathcal{R}$ is approximated by a variational distribution $Q(\mathcal{R})$ such that $P(\mathcal{R}|\mathcal{E}) \approx Q(\mathcal{R})$. In our setting, $P(\mathcal{R}|\mathcal{E})$ is the *conditioned latent representation* $\mathcal{R}_C$, obtained as a result of the encoding stage. This distribution $Q(\mathcal{R})$ is restricted to a family of distributions simpler than $P(\mathcal{R}|\mathcal{E})$, and inference if performed by selecting the distribution that minimizes a dissimilarity function $D(Q||P)$. Following standard practice, we use the Kullback-Leibler (KL) divergence [38] of Q from P as the dissimilarity function:

$$D_{KL}(Q||P) \triangleq \sum_{\mathcal{R}} Q(\mathcal{R}) \log \frac{Q(\mathcal{R})}{P(\mathcal{R}|\mathcal{E})} \tag{1}$$

Practically, this is achieved by doubling the dimensionality of $\mathcal{R}$ to $N_l \times 2D_l$, with each half storing respectively the mean μ_l and standard deviation σ_l of the variational distribution. After conditioning $\mathcal{R}_C$ on input embeddings $\mathcal{E} = \mathcal{E}_I \oplus \mathcal{E}_G$, a $N_l \times D_l$ *sampled latent representation*

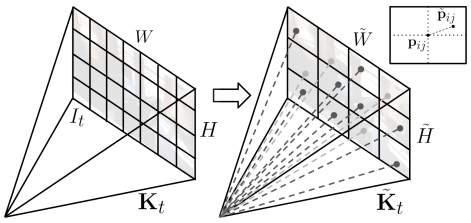


Figure 3: **Example of encoder-level data augmentation**. The input image I_t is resized from resolution 4×7 to 3×4, with the corresponding change in $\mathbf{K}_t$ to preserve 3D properties (Equation 3). The 2D location of each pixel $\mathbf{p}_{ij}$ is perturbed and used to generate geometric embeddings $\mathcal{E}_G$. Finally, a percentage of embeddings is discarded.

$\mathcal{R}_S$ is generated by sampling from $\mathcal{N}(\mu_c, \sigma_c)$, and can be decoded to generated depth predictions. During training, a single sample is generated, and an additional KL divergence loss regularizes our variational distribution (Equation 8). During inference, multiple samples $\{\mathcal{R}_S^n\}_{n=1}^N$ can be generated from the same $\mathcal{R}_C$, each leading to a different decoded depth map $\tilde{D}_n$. We show that these predictions statistically approximate per-pixel depth uncertainty, and can be used to improve performance by selectively removing pixels with high uncertainty values (Figure 7). Each pixel $\mathbf{p}_{ij}$ has a mean μ_{ij} and standard deviation σ_{ij} given by:

$$\mu_{ij} = \frac{1}{N}\sum_N \tilde{d}_{ij}^n \qquad \sigma_{ij} = \sqrt{\frac{\sum_N (\tilde{d}_{ij}^n - \mu_{ij})^2}{N}} \tag{2}$$

3.5. Encoder-Level Data Augmentation

Differently from traditional architectures for monocular depth estimation [19, 23, 50, 39], ZeroDepth follows [28] and decouples the encoding and decoding stages, which enables the decoding of estimates from embeddings that were not encoded. We take advantage of this property to decode estimates using only geometric embeddings, and empirically show that this leads to improvements over standard encoder-decoder architectures. In [28] a series of decoder-level geometry-preserving augmentations was proposed, leading to increased viewpoint diversity for multi-view depth estimation. Alternatively, here we introduce a series of *encoder-level* data augmentation techniques, designed to improve robustness to appearance and geometric domain gaps (see Figure 3). Note that decoder information, i.e. the geometric embeddings used as queries and the depth maps used as supervision, is not modified in any way.

Resolution Jittering. Our geometric embeddings are generated given pixel coordinates $\mathbf{p}_{ij}$ and camera intrinsics $\mathbf{K}_t$, and therefore are invariant to image resolution (assuming that $\mathbf{K}_t$ is scaled accordingly). However, this is not the case for image embeddings, that are appearance-based with fixed receptive fields, and therefore will change depending on image resolution. Moreover, since our focus is direct transfer, there is no guarantee that test images will have the same resolution as training images. Because of that, we randomly resize images during training, from $H \times W$ to $\tilde{H} \times \tilde{W}$, thus modifying the CNN features used as image embeddings for encoding. The 3D scene structure (including metric scale) is preserved by also modifying camera intrinsics, such that:

$$\tilde{\mathbf{K}}_t = \begin{bmatrix} r_w f_x & 0 & r_w(c_x - 0.5) + 0.5 \\ 0 & r_h f_y & r_h(c_y - 0.5) + 0.5 \\ 0 & & 1 \end{bmatrix} \tag{3}$$

where $r_w = \tilde{W}/W$ and $r_h = \tilde{H}/H$ are respectively the width and height resizing ratios.

Ray Jittering. Geometric embeddings are calculated using 2D pixel coordinates $\mathbf{p}_{ij}$, located at their center. Because of that, images with the same resolution and intrinsics will always generate the same geometric embeddings. Moreover, since image size is discrete, resolution jittering as described previously will not produce a continuous distribution of viewing rays that covers the entire operational space. To ensure a proper coverage, we also perturb $\mathbf{p}_{ij}$ by injecting uniform noise between $[-0.5, 0.5]$, such that the new location $\tilde{\mathbf{p}}_{ij}$ is still within the pixel boundaries. This simple modification promotes a larger diversity of geometric embeddings during training, and by extension facilitates transfer to different resolutions and camera geometries. Corresponding image embeddings $\mathcal{E}_I$ are generated by bilinearly interpolating image features in these new coordinates.

Embedding Dropout. At training time we randomly drop a proportion p of the encoder embeddings, where p is uniformly sampled from $[0, 0.5]$. This dropout regularization effectively promotes the learning of more robust latent representations, by encouraging the model to reason over and generate dense predictions conditioned on sparse input.

3.6. Training Losses

Our training objective has three components: depth supervision, surface normal regularization, and KL divergence, each with its own weight coefficient (for simplicity, we assume $\alpha_D = 1$). Below we describe each one in detail.

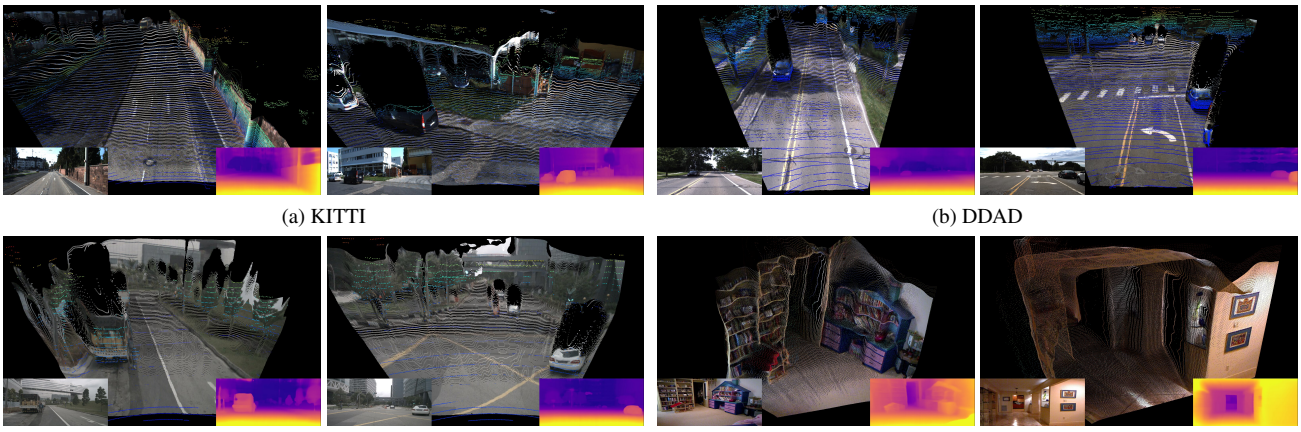
$$\mathcal{L} = \mathcal{L}_D + \alpha_N \mathcal{L}_N + \alpha_K \mathcal{L}_K \tag{4}$$

Depth Supervision. We use a *smooth L1 loss* to supervise depth predictions $\hat{D}_t$ relative to groundtruth depth maps D_t. Assuming $\Delta d_{ij} = |d_{ij} - \hat{d}_{ij}|$ to be the pixel-wise absolute depth error, it is defined as:

$$\mathcal{L}_D = \frac{1}{N}\sum_{ij \in D_t} \begin{cases} 0.5 * \Delta d^2/\beta & \text{if } \Delta d < \beta \\ \Delta d - 0.5 * \beta & \text{otherwise} \end{cases} \tag{5}$$

where N is the number of valid pixels $\mathbf{p}_{ij} = (u, v)$ in D_t, and β is a threshold for the change between losses.

Surface Normal Regularization. As additional regularization, we follow [26] and leverage the dense labels from



(a) KITTI (b) DDAD

(c) nuScenes (d) NYUv2

Figure 4: **ZeroDepth qualitative zero-shot depth estimation results using the same pre-trained model.** We overlay colored predicted monocular pointclouds with ground-truth pointclouds shown as height maps. Our framework is capable of zero-shot metric depth estimation across datasets with different camera geometries and depth ranges.

synthetic datasets to also minimize the error between normal vectors produced by groundtruth and predicted depth maps. For a pixel $\mathbf{p}$, its normal vector $\mathbf{n} \in \mathbb{R}^3$ is defined as:

$$\mathbf{n} = \left(\mathbf{P}_{u+1,v} - \mathbf{P}_{u,v}\right) \times \left(\mathbf{P}_{u,v+1} - \mathbf{P}_{u,v}\right) \tag{6}$$

where $\mathbf{P}_{ij} = (x, y, z) = d_{ij}\, \mathbf{K}_t^{-1}\, [u, v, 1]^T$ is the *unprojection* of $\mathbf{p}$ into 3D space. As a measure of proximity between vectors, we use the *cosine similarity* metric:

$$\mathcal{L}_N = \frac{1}{2N} \sum_{\mathbf{p} \in D} \left(1 - \frac{\hat{\mathbf{n}} \cdot \mathbf{n}}{||\hat{\mathbf{n}}||\, ||\mathbf{n}||}\right) \tag{7}$$

KL Divergence. We also minimize the Kullback-Leibler (KL) divergence of our variational latent representation, which promotes the learning of a Gaussian distribution that is sampled during the decoding stage:

$$\mathcal{L}_{KL} = -\frac{1}{2N} \sum_{ij \in D_t} 1 + s_{ij} - \mu_{ij}^2 - \exp(s_{ij}) \tag{8}$$

where μ is the mean and $s = \log \sigma^2$ is the log-variance of our conditioned latent representation (Section 3.4).

4. Experiments

4.1. Training Datasets

Parallel Domain [25, 26]. The Parallel Domain dataset is procedurally generated, with photo-realistic renderings of urban driving scenes. We use the splits from [25] and [26], containing 40000 and 52500 images from 6 cameras.
TartanAir [59]. We use the TartanAir dataset as an additional source of synthetic data and camera geometries. It contains a total of 306637 images from 2 stereo cameras.
Waymo [53]. The Waymo dataset is our primary source of real-world training data. We use the official training and validation splits, for a total of 198068 images from 5 cameras, with LiDAR groundtruth.
Large-Scale Driving (LSD). As an additional source of real-world training data, we collected depth-annotated images using in-house vehicles (further details are omitted for anonymity). It contains a total of 176320 images from 6 cameras, with LiDAR groundtruth.
OmniData [10]. The OmniData dataset is composed of a collection of synthetic datasets. For our indoor experiments, we used a combination of the Taskonomy, HM3D, Replica, and Replica-GSO splits, for a total of 14340580 images.

4.2. Evaluation Datasets

KITTI [16]. The KITTI dataset is the standard benchmark for depth estimation. We evaluate on the *Eigen* split [11], composed of 697 images. Following standard protocol, we consider distances up to 80m and use the *garg* crop [15].
DDAD [23]. The DDAD dataset includes multiple cameras and long-range sensors for ground-truth depth maps. We use the official validation split, with 3950 images from 6 cameras, and consider distances up to 200m without crops.
nuScenes [7]. The nuScenes dataset is a well-known benchmark for multi-camera 3D object detection. We use the official validation split, with 6019 images from 6 cameras, and consider distances up to 200m without crops.
NYUv2 [48]. The NYUv2 dataset is a widely used benchmark for indoor monocular depth estimation. We use the official validation split, with 654 images and groundtruth depth maps captured by a Kinect RGB-D camera.

4.3. Scale-Aware Monocular Depth Estimation

We evaluated the zero-shot metric scale transfer capabilities of ZeroDepth across multiple traditional depth estimation benchmarks, both *indoors* and *outdoors* (Figure 4). To this end, we trained a single model using a combination of

Method	Supervision	Med. Scale	KITTI						
			Lower is better				*Higher is better*		
			AbsRel	SqRel	RMSE	RMSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 [19]	M	✓	0.115	0.903	4.863	0.193	0.877	0.959	0.981
PackNet-SfM [23]	M	✓	0.111	0.785	4.601	0.189	0.878	0.960	0.982
	M+v	✗	0.111	0.829	4.788	0.199	0.864	0.954	0.980
GUDA [26]	M+Sem	✓	0.107	0.714	4.421	—	0.883	—	—
MonoDEVSNet [29]	M	✓	0.104	0.721	4.396	0.185	0.880	0.962	0.983
FeatDepth [52]	M	✓	0.104	0.729	4.481	0.179	0.893	0.965	0.982
Guizilini *et al.* [24]	M+Sem	✓	0.102	0.698	4.381	0.178	0.896	0.964	0.984
Chawla *et al.* [8]	M+GPS	✓	0.112	0.894	4.852	0.192	0.877	0.958	0.981
		✗	0.109	0.860	4.855	0.198	0.865	0.954	0.980
Swami *et al.* [54]	M+V	✓	0.103	0.654	4.300	0.178	0.891	0.966	0.984
		✗	0.109	0.702	4.409	0.185	0.876	0.962	0.984
ZeroDepth (outdoor)	—	✓	0.102	**0.627**	**4.044**	**0.172**	**0.910**	**0.980**	**0.996**
		✗	**0.100**	0.662	4.213	0.181	0.899	0.973	0.992
ZeroDepth	—	✓	0.105	0.647	4.194	0.178	0.886	0.965	0.984
		✗	0.102	0.728	4.378	0.196	0.892	0.961	0.977

Method	Supervision	Med. Scale	*DDAD (all)*			*nuScenes*		
			AbsRel↓	RMSE↓	$\delta < 1.25$↑	AbsRel↓	RMSE↓	$\delta < 1.25$↑
Monodepth2 [19]	M	✓	0.217	12.962	0.699	0.287	7.184	0.641
PackNet-SfM [23]	M	✓	0.234	13.253	0.672	0.309	7.994	0.547
ZoeDepth* [4]	M	✗	0.647	16.320	0.265	0.504	7.717	0.255
FSM [27]	M+e	✓	0.203	12.810	0.716	0.301	7.892	0.729
		✗	0.205	13.688	0.672	0.319	7.860	0.716
VolumetricFusion [37]	M+e	✓	0.221	13.031	0.681	0.271	7.391	0.726
		✗	0.218	13.327	0.674	0.289	7.551	0.709
SurroundDepth [60]	M+e	✓	0.200	12.270	0.740	0.245	**6.835**	0.719
		✗	0.208	12.977	0.693	0.280	7.467	0.661
ZeroDepth (outdoor)	—	✓	0.160	10.814	0.811	0.236	7.054	0.747
		✗	0.161	11.034	0.813	0.255	7.205	0.746
ZeroDepth	—	✓	**0.156**	**10.678**	**0.814**	**0.221**	7.226	**0.754**
		✗	0.157	10.818	0.810	0.234	7.485	0.717

Table 1: **Depth estimation results on KITTI [16], DDAD [23], and nuScenes [7].** *Supervision* refers to the training supervision used in the target dataset (*M* for monocular self-supervision, *v* for velocity, *V* for synthetic data with similar camera geometry, *e* for extrinsics, and *Sem* for semantic segmentation), and *Med. Scale* refers to the use of ground-truth median-scaling during evaluation. Best and second best overall numbers are **bolded** and underlined. Best and second best median-scaled numbers are colored in shades of blue, and best and second best metric numbers are colored in shades of red. Methods with * were obtained using the official pre-trained model, evaluated following the standard protocol for each dataset.

the *Parallel Domain*, *TartanAir*, *Waymo*, and *LSD* outdoor datasets, with a total of $3,159,424$ samples, as well as the *Omnidata* indoor dataset, with a total of $14,340,580$ samples. Outdoor samples were repeated 5 times, to ensure a similar distribution to indoor samples, resulting in a total of $30,137,700$ samples. A single depth decoder was used, with a maximum range of 200m. The training session was distributed across 8 A100 GPUs, with a batch size $b = 16$ per GPU, for 10 epochs, totalling roughly 7 days (for additional details, please refer to the supplementary material).

This model was then evaluated on the *KITTI*, *DDAD*, *nuScenes*, and *NYUv2* datasets *without fine-tuning*, using the standard evaluation protocol for each benchmark. Quantitative results for all these datasets are shown in Tables 1 and 2. Due to a lack of baselines capable of zero-shot transfer for comparison, we also included methods that (i) self-supervise on the target dataset, using temporal context frames; and (ii) rely on median-scaling at test time to generate metric predictions. As we can see, ZeroDepth outperforms all published methods, despite never having seen any of the target data. Compared to other methods that predict metric depth, on the KITTI dataset, ZeroDepth significantly improves upon methods that use velocity as weak supervision [23], as well as synthetic data with similar camera geometry [54] and GPS information [8]. On the DDAD and nuScenes datasets, ZeroDepth also outperforms vari-

Method	Supervision	Med. Scale	Lower is better		Higher is better	
			AbsRel	RMSE	$\delta < 1.25$	$\delta < 1.25^2$
Monodepth2 [19]	M	✓	0.160	0.601	0.767	0.949
SC-Depth [19]	M	✓	0.159	0.608	0.772	0.939
P^{2}Net (5-frame)	M	✓	0.147	0.553	0.801	0.951
Bian *et al.* [5]	M	✓	0.147	0.536	0.804	0.950
Struct2Depth [43]	M	✓	0.142	0.540	0.813	0.954
MonoIndoor [35]	M	✓	0.134	0.526	0.823	0.958
MonoIndoor++ [44]	M	✓	0.132	0.517	0.834	0.961
DistDepth [61]	—	✓	0.158	0.548	0.791	0.942
DPT + OmniData [10]	—	✓	0.089	0.348	0.921	0.981
ZeroDepth (indoor)	—	✓	0.084	**0.321**	0.921	0.983
ZeroDepth	—	✓	**0.081**	0.338	**0.926**	**0.986**
DistDepth [61]	—	✗	0.289	1.077	0.706	0.934
ZeroDepth (indoor)	—	✗	0.104	0.389	0.895	0.965
ZeroDepth	—	✗	0.100	0.380	0.901	0.961

Table 2: **Depth estimation results on the NYUv2 [48] dataset.** ZeroDepth outperforms published methods that use self-supervision in the target dataset (M) and median-scaling during evaluation (*Med. Scale*), and improves upon [10] by enabling the transfer of metric scale across datasets.

ous methods that rely on cross-camera extrinsics to recover metric scale [27, 37, 60]. A similar trend is observed on the NYUv2 dataset, where ZeroDepth outperforms several methods that rely on in-domain self-supervision, as well as test-time median-scaling. The only method that is competitive to ours in terms of median-scaled evaluation is [10], that uses DPT [50] pre-trained on the same dataset. However, we note that this method uses a scale-invariant loss, and thus abstracts away geometry to focus solely on appearance features. Because of that, it is unable to generate metric predictions, whereas our method achieves similar median-scaled performance while establishing a new state-of-the-art in zero-shot metric depth estimation.

We also evaluated ZeroDepth variants trained specifically for each setting (indoors and outdoors). The outdoor model was trained using the *Parallel Domain*, *TartanAir*, *Waymo*, and *LSD* datasets, for 20 epochs, in roughly 4.5 days. The outdoor model was trained using the *Omnidata* dataset, for 5 epochs, in roughly 4.5 days as well. As we can see, the generalization to both settings did not impact performance in a meaningful way, leading only to marginally worse outdoor results, and actually improved indoor results. We also note that, differently from [4], our entire model is reutilized across settings, without specialized adaptive decoders for different depth ranges.

4.4. Ablative Analysis

Here we discuss the various components of ZeroDepth, analyzing design choices and robustness to different parameter choices. Additional ablations considering other aspects can be found in the supplementary material.

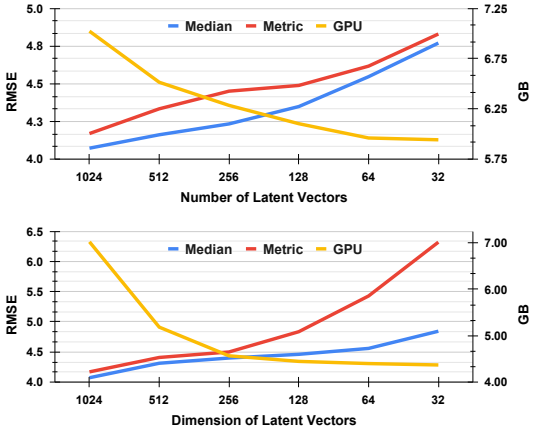


Figure 5: **Network complexity ablation** on the KITTI dataset, for different latent space sizes with the GPU inference requirements to process a 192×640 image.

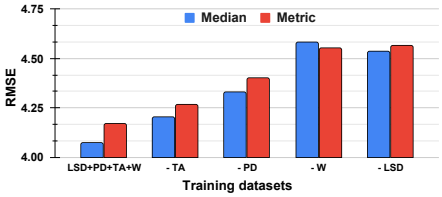


Figure 6: **Effects of removing individual datasets** on zero-shot transfer results to the KITTI dataset.

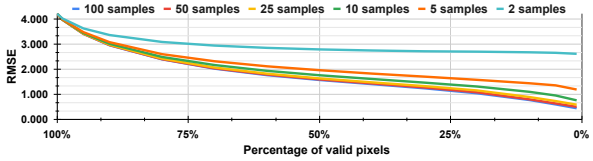


Figure 7: **Depth estimation performance on KITTI** with varying confidence levels. Valid pixels are selected filtered on the standard deviation from multiple samples.

Network Complexity. The first component we ablate (Figure 5) is the size of our latent representation $\mathcal{R}$, in terms of number N_l and dimension D_l of latent vectors. As expected, reducing $\mathcal{R}$ leads to a steady degradation in results. In particular, reducing N_l leads to a roughly linear degradation, although even with $N_l = 32$ we still achieve performance comparable with monodepth2 [19] (RMSE 4.881 v. 4.863). Interestingly, reducing D_l leads to a much faster degradation in metric results (at $D_l = 32$ we observe an RMSE of 4.904 for median-scaled and 6.421 for metric results). This is evidence that our model is not simply learning to produce metrically scaled predictions from depth supervision, but rather additional scale priors that can be transferred across datasets. As we decrease network complexity, the model is unable to properly learn these priors, and hence metric predictions degrade at a faster rate.

Training Datasets. We also ablate the impact of different training datasets in the final performance. In Figure 6 we

Evaluation	Dataset	*Lower is better*				*Higher is better*		
		AbsRel	SqRel	RMSE	RMSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
KITTI	PWA [42]	0.060	0.221	2.604	0.093	0.958	0.994	**0.999**
	BTS [40]	0.059	0.245	2.756	0.096	0.956	0.993	0.998
	AdaBins [3]	0.058	0.190	2.360	0.088	0.964	**0.995**	**0.999**
	ZoeDepth [4]	0.057	0.194	2.290	0.091	0.967	**0.995**	**0.999**
	ZeroDepth	**0.053**	**0.164**	**2.087**	**0.083**	**0.968**	**0.995**	**0.999**
DDAD (front)	BTS [40]	0.094	1.913	11.437	0.212	0.888	0.947	0.978
	PackNet [21]	0.088	1.760	11.331	0.195	0.899	0.960	0.981
	ZeroDepth	**0.086**	**1.709**	**10.652**	**0.180**	**0.909**	**0.967**	**0.984**
DDAD (all)	BTS [40]	0.153	2.413	10.437	0.272	0.813	0.915	0.956
	PackNet [21]	0.145	2.318	10.049	0.242	0.845	0.909	0.951
	ZeroDepth	**0.142**	**2.234**	**9.842**	**0.235**	**0.851**	**0.939**	**0.967**
nuScenes	**ZeroDepth**	**0.143**	**1.508**	**4.891**	**0.233**	**0.862**	**0.938**	**0.966**
NYUv2	BinsFormer [45]	0.094	—	0.330	0.040	0.925	0.989	0.997
	PixelFormer [1]	0.090	—	0.322	0.039	0.929	0.991	0.998
	VA-Depth [46]	0.086	—	0.304	0.036	0.939	0.992	0.999
	ZoeDepth [4]	0.077	—	0.277	0.033	0.953	0.995	0.999
	ZeroDepth	**0.074**	**0.031**	**0.269**	**0.103**	**0.954**	**0.995**	**1.000**

Table 3: **In-domain depth estimation results**, obtained by fine-tuning ZeroDepth on the training splits of each evaluation dataset. Best and second best numbers are **bolded** and underlined. Reported results are metric (i.e., without median-scaling).

show results when removing each of the 4 outdoor datasets, and training for the same number of iterations. As expected, removing each individual dataset results in some amount of degradation, with median-scaled and metric results degrading by roughly the same amount. Removing real-world datasets (*LSD* and *Waymo*) has the highest impact in overall performance, both because these datasets are larger and also because they decrease the appearance domain gap between training and evaluation datasets. Complete tables are available in the supplementary material.

Variational Uncertainty. Here we evaluate the quality of uncertainty estimates generated by our variational architecture. In Figure 7 we show results using different percentages of valid depth pixels, filtered from lowest to highest standard deviation based on a varying number of samples. We see a steady performance increase as pixels with lower standard deviation are considered, indicating that our variational architecture succeeds in detecting areas with higher uncertainty. Moreover, as we increase the sample size the quality of the estimated distribution also increases, leading to further improvements until saturation at around 50 samples. At this point, selecting the top 50% pixels leads to a 58% improvement, from RMSE 4.044 to 1.678. Qualitative results are shown in the supplementary material.

4.5. Fine-Tuned Depth Estimation

Even though ZeroDepth is designed for the zero-shot setting (i.e., a single model is trained and directly evaluated on other datasets), if in-domain data is available it is possible to fine-tune our original model to further improve performance for a specific dataset. Here we explore this capability and fine-tune ZeroDepth in each of the evaluation datasets. We start from the same pre-trained weights, and for each dataset we train for 5 additional epochs on its corresponding training split, with a learning rate of $lr = 10^{-5}$. The evaluation procedure is the same, except for KITTI, where we use instead the split proposed in [55] because it is the standard protocol reported by supervised methods. Quantitative results are reported in Table 3, and show that fine-tuning ZeroDepth with in-domain data leads to significant improvements, to the point of outperforming state-of-the-art methods trained specifically for each dataset.

5. Conclusion

In this paper we introduce ZeroDepth, a novel monocular depth estimation architecture that enables the robust zero-shot transfer of metric scale across datasets, via large-scale supervised training to learn scale priors from a combination of image and geometric embeddings. We maintain a global variational latent representation, that is conditioned using information from a single frame during the encoding stage, and can be sampled and decoded to generate multiple depth maps in a probabilistic fashion. We also propose a series of encoder-level data augmentation techniques, designed to address the appearance and geometric domain gaps between datasets collected in different locations with different cameras. We evaluated the same pre-trained ZeroDepth model on both indoor and outdoor settings, and demonstrated state-of-the-art results in multiple benchmarks, outperforming methods that rely on in-domain self-supervision and test-time median-scaling.

A. Training Details

We implemented our models using PyTorch [49], with distributed training across 8 A100 GPUs and TensorFloat-32 precision format. We use the AdamW optimizer [47], with standard parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, a weight decay of $w = 10^{-4}$, batch size of $b = 16$, and an initial learning rate of $lr_1 = 10^{-4}$. During the first epoch, we linearly warm the learning rate up from $lr_0 = 10^{-5}$. Afterwards, we decay the learning rate by a factor of $\gamma = 0.8$ after every 5 epochs for outdoor experiments, and 2 epochs for indoor experiments, such that $lr_{n+1} = \gamma lr_n$. In addition to our proposed encoder-level data augmentation techniques, we also apply random horizontal flipping with 50% probability, and color jittering of $(0.5, 0.5, 0.5, 0.1)$ respectively for brightness, contrast, saturation and hue.

For resolution jittering, we randomly resize input images to resolutions between 25% and 150% of the original $H \times W$, independently for the height and width dimensions. Due to network architecture restrictions, we round up our sampled resolutions to be multiples of 32. For embedding dropout, we randomly select a number of encoder embeddings between 0% and 50% to remove at each training iteration. During evaluation we do not perform any sort of data augmentation. For the loss calculation, we multiply the surface normal regularization term by $\alpha_N = 0.2$, and the KL-divergence term by $\alpha_{KL} = 0.1$. To decrease memory requirements and computational complexity, during training we use strided ray sampling [34] to downsample the decoded image to $1/8$ the original resolution.

B. Network Architecture

We use a ResNet18 [30] backbone as the encoder to generate 960-dimensional image embeddings. Our geometric embeddings are calculated using $F = 16$ frequency bands and $\mu = 64$ as the maximum resolution, resulting in 51-dimensional vectors. Our latent representation is of dimensionality 1024×1024, with 8 self-attention heads and 8 self-attention layers for conditioning, including GeLU activations [31] and dropout of 0.1. We use a single cross-attention layer for conditioning, and another single cross-attention layer for decoding, followed by an MLP that projects the output to a 1-dimensional depth estimate. For uncertainty estimation, we decode 10 depth maps, from different sampled latent representations, and calculate the pixel-level mean μ_{ij} and standard deviations σ_{ij}. In total, ZeroDepth has $232,591,380$ parameters.

C. Extended Depth Estimation Tables

For completeness, in Tables 5 and 6 we provide depth estimation results for each individual camera of the *DDAD* and *nuScenes* datasets. These results are obtained using the outdoor variant of ZeroDepth, and were averaged to generate our entries in Tables 1 and 2 of the main paper. Moreover, in Table 7 we report the full depth estimation results of our ablation regarding the use of different training datasets (see Figure 6 of the main paper, where due to space constraints we only report *KITTI* results). In these results we observe a similar trend: performance consistently degrades across all evaluation datasets as we consider fewer training datasets, and the degradation is similar between metric and median-scaled predictions.

	Method	Med. Scale	*Lower is better*		*Higher is better*	
			AbsRel	RMSE	$\delta < 1.25$	$\delta < 1.25^2$
A	ResNet18 [19]	✓	0.194	5.489	0.718	0.916
		✗	0.233	5.961	0.639	0.869
B	ResNet50 [19]	✓	0.191	5.530	0.713	0.903
		✗	0.224	5.885	0.632	0.868
C	DPT [50]	✓	0.126	4.615	0.863	0.971
		✗	0.240	6.090	0.590	0.936
D	DPT w/ Geom. Embed.	✓	0.119	4.287	0.869	0.966
		✗	0.144	4.712	0.808	0.956
E	w/o Geom. Embed.	✓	0.135	4.818	0.819	0.961
		✗	0.175	5.332	0.751	0.923
F	w/o Surface Normal	✓	0.107	4.277	0.881	0.978
		✗	0.109	4.379	0.879	0.977
G	w/o Variational $\mathcal{R}$	✓	0.113	4.461	0.877	0.979
		✗	0.122	4.702	0.869	0.973
H	w/o Res. Jittering	✓	0.126	4.666	0.865	0.969
		✗	0.142	4.884	0.824	0.963
I	w/o Ray Jittering	✓	0.105	4.177	0.898	0.976
		✗	0.112	4.598	0.881	0.972
J	w/o Embed. Dropout	✓	0.104	4.129	0.901	0.974
		✗	0.109	4.432	0.887	0.971
	ZeroDepth	✓	0.102	**4.044**	**0.910**	**0.980**
		✗	**0.100**	4.213	0.899	0.973

Table 4: **ZeroDepth ablation study** on the KITTI dataset.

In particular, improvements seem to be correlated with the number of training tokens available on each dataset: considering 384×640 resolution images, and an encoding downsample ratio of 4 (Section 3.3, main paper), each image contains a total of 15360 tokens. Therefore, the *PD* dataset has roughly 8.5B tokens, followed by *TartanAir* with 9.4B, *Waymo* with 1.5T, and *LSD* with 1.6T tokens. Note that this is without considering our proposed encoder-level data augmentation techniques (Section 3.5, main paper), that further increases training token diversity by (i) modifying the CNN features used as image embeddings; and (ii) perturbing the geometric embeddings to cover the entire camera field of view. Increasing the number of training tokens by ingesting additional datasets, as well as increasing network complexity to enable proper learning from such diverse data, are straightforward ways to further increase performance within our framework.

D. Variational Uncertainty Sampling

In Figure 9 we show an example of predicted variational uncertainty, and how it can be used to improve depth estimation by selecting pixels with higher confidence levels. As expected (Figure 9a), uncertainty increases with longer ranges, and is also larger in areas with sudden depth discontinuities (i.e., object boundaries), that are usually smoothed out to generate a characteristic "bleeding" effect across modes. By removing as few as 10% of the valid depth pixels, we already observe a significant improvement of 30% in Root Mean Squared Error (RMSE), from 4.044 to 2.859, mostly due to the removal of areas with bleeding artifacts. In fact, the overall pointcloud structure (i.e., observed cars, ground plane and walls) is preserved even when we remove as much as 50% of valid depth pixels, leading to an RMSE improvement of 63% relative to the full pointcloud.

E. Full Surround Pointclouds

The *DDAD* and *nuScenes* datasets have multiple cameras in each sample, which enables the reconstruction of full surround pointclouds by combining reconstructions from each individual camera. This property has been explored in several works [27, 60], as a way to generate scale-aware depth maps by exploiting cross-camera extrinsics as a source of metric information. In Fig 10 we show examples of ZeroDepth pointclouds for each of these datasets, obtained by overlaying individual pointclouds from the 6 cameras in a single sample. We emphasize that these are direct transfer results, generated by evaluating ZeroDepth without fine-tuning, and these are *single-frame* results, meaning that each image was processed independently, and the reconstructed pointclouds were combined without any post-processing or alignment procedure. As we can see, these individual pointclouds seamlessly blend in overlapping areas, which indicates that our *learned scale is consistent across multi-cameras*, including across cameras with different intrinsics, resolutions, and relative vehicle orientation. Furthermore, as shown by the LiDAR pointclouds overlaid with the pointclouds, our learned scale is not only consistent across cameras, but *it is also metric*, i.e. it aligns with the "ground-truth" LiDAR information without any required post-processing.

F. Additional Ablative Analysis

Network Architecture. In Table 4 we ablate the different components of ZeroDepth, starting with the choice of network architecture. To that end, we trained under the same conditions (including augmentations) both Monodepth2 [19] models with ResNet backbones, as an example of CNN-based networks, as well as a DPT [50] model, as an example of Transformer-based network without an intermediate latent representation. As shown, Monodepth2 models struggle both in terms of median-scaled and metric predictions (**A** and **B**), regardless of network complexity. The DPT model (**C**) achieves better median-scaled performance, however it still struggled to transfer scale across datasets. To test our first claim, that input-level geometric information is key to scale transfer, we modified DPT to include the same geometric embeddings used in ZeroDepth. Interestingly, this simple modification (**D**) not only improved median-scaled performance (0.126 to 0.119 AbsRel), but also significantly impacted metric performance (0.240 to 0.144). Even so, ZeroDepth still outperforms this DPT variant by a large amount (0.100 vs. 0.144). This is evidence of our second claim, that maintaining an intermediate latent representation is beneficial for scale transfer.

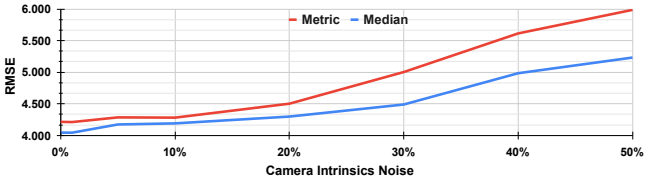


Figure 8: **Depth estimation performance on KITTI** with different noise levels for camera intrinsics.

Design Choices. We also ablate in Table 4 the different design choices of ZeroDepth. Firstly, we show that replacing our 3D geometric embeddings with 2D positional embeddings (**E**) leads to a large degradation in metric performance. This is in accordance with our previous DPT experiments, however even in such conditions ZeroDepth still outperforms DPT by a large margin (0.175 vs. 0.240 AbsRel). We also removed the surface normal regularization term (**F**), and observed some amount of degradation, showing that dense synthetic data can be leveraged for additional structural supervision. Afterwards, we experimented with replacing our variational latent representation (**G**) with the original representation from [33], as well as removing our various encoder-level data augmentations, namely (**H**) resolution jittering, (**I**) ray jittering, and (**J**) embedding dropout. Each component contributes to improvements in depth estimation across datasets, particularly for metric predictions.

Camera Intrinsics. Although our framework does not require camera poses, it still requires intrinsic calibration. In Figure 8 we ablate the impact of perturbing samples during evaluation, by adding random noise to their camera parameters. As expected, performance degrades with higher noise levels, since geometric embeddings become increasingly inaccurate. Moreover, we observe a much steeper degradation in metric predictions, relative to median-scaled ones. This is further evidence that our model goes beyond simply generating metric predictions, and relies instead on learned scale priors based on physical properties. These scale priors require accurate intrinsics to correlate 2D information with 3D properties, which leads to degradation when camera parameters are inaccurate.

Method	Camera	Med.Scale	*Lower is better*				*Higher is better*		
			AbsRel	SqRel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
ZeroDepth	Front	✗	0.100	1.916	11.214	0.188	0.895	0.962	0.983
	Front-Left		0.148	2.245	10.011	0.249	0.833	0.932	0.965
	Front-Right		0.182	2.934	10.397	0.286	0.771	0.908	0.951
	Back-Left		0.165	2.642	10.648	0.269	0.806	0.918	0.957
	Back-Right		0.205	3.268	10.484	0.309	0.748	0.893	0.969
	Back		0.157	2.656	12.135	0.248	0.813	0.933	0.969
ZeroDepth	Front	✓	0.100	1.950	11.318	0.191	0.889	0.961	0.982
	Front-Left		0.151	2.325	10.067	0.254	0.818	0.931	0.965
	Front-Right		0.179	3.113	10.874	0.308	0.760	0.893	0.941
	Back-Left		0.170	2.555	10.728	0.279	0.782	0.912	0.955
	Back-Right		0.206	3.053	10.591	0.332	0.714	0.875	0.930
	Back		0.159	2.806	12.627	0.265	0.808	0.917	0.962

Table 5: **Per-camera ZeroDepth depth estimation results** on the DDAD [23] dataset.

Method	Camera	Med.Scale	*Lower is better*				*Higher is better*		
			AbsRel	SqRel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
ZeroDepth	Front	✗	0.150	2.101	7.484	0.240	0.839	0.939	0.969
	Front-Left		0.287	4.931	7.300	0.363	0.711	0.862	0.920
	Front-Right		0.420	12.247	7.545	0.391	0.690	0.853	0.913
	Back-Left		0.193	3.615	7.818	0.291	0.796	0.910	0.952
	Back-Right		0.252	2.970	6.411	0.340	0.709	0.866	0.924
	Back		0.226	2.516	6.669	0.331	0.732	0.881	0.932
ZeroDepth	Front	✓	0.157	2.154	7.612	0.239	0.822	0.941	0.971
	Front-Left		0.259	3.913	7.063	0.341	0.716	0.876	0.929
	Front-Right		0.354	6.899	7.043	0.365	0.690	0.851	0.920
	Back-Left		0.192	3.095	7.639	0.281	0.789	0.917	0.958
	Back-Right		0.230	2.728	6.275	0.321	0.735	0.878	0.930
	Back		0.223	2.609	6.693	0.317	0.731	0.883	0.935

Table 6: **Per-camera ZeroDepth depth estimation results** on the nuScenes [7] dataset.

References

[1] Ashutosh Agarwal and Chetan Arora. Attention attention everywhere: Monocular depth prediction with skip attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5861–5870, January 2023. 8

[2] Manuel López Antequera, Pau Gargallo, Markus Hofinger, Samuel Rota Bulò, Yubin Kuang, and Peter Kontschieder. Mapillary planet-scale depth dataset. In *European Conference on Computer Vision*, pages 589–604. Springer, 2020. 1, 2

[3] Farooq Shariq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 8

[4] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth, 2023. 2, 6, 7, 8

[5] Jiawang Bian, Huangying Zhan, Naiyan Wang, Tat-Jun Chin, Chunhua Shen, and Ian D. Reid. Unsupervised depth learning in challenging indoor video: Weak rectification to rescue. *ArXiv*, abs/2006.02708, 2020. 7

[6] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. 3

[7] 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 *CVPR*, 2020. 5, 6, 11

[8] H. Chawla, A. Varma, E. Arani, and B. Zonooz. Multi-

Evaluation	Dataset	Med. Scale	*Lower is better*				*Higher is better*		
			AbsRel	SqRel	RMSE	RMSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
KITTI	– TA	✓	0.104	0.651	4.011	0.174	0.905	0.978	0.995
		✗	0.103	0.670	4.171	0.187	0.891	0.970	0.991
	– PD	✓	0.109	0.697	4.227	0.179	0.899	0.977	0.995
		✗	0.105	0.720	4.400	0.192	0.886	0.968	0.990
	– W	✓	0.118	0.858	4.579	0.188	0.881	0.974	0.993
		✗	0.110	0.831	4.552	0.199	0.876	0.962	0.988
	– LSD	✓	0.121	0.753	4.536	0.198	0.872	0.968	0.991
		✗	0.133	0.830	4.562	0.207	0.861	0.963	0.988
	All	✓	0.102	0.627	4.044	0.172	0.910	0.980	0.996
		✗	0.100	0.662	4.213	0.181	0.899	0.973	0.992
DDAD	– TA	✓	0.166	2.889	11.576	0.284	0.808	0.908	0.953
		✗	0.168	2.927	11.744	0.294	0.791	0.901	0.950
	– PD	✓	0.181	2.954	11.988	0.283	0.784	0.902	0.951
		✗	0.183	3.025	12.238	0.295	0.774	0.893	0.957
	– W	✓	0.198	3.470	12.767	0.328	0.772	0.886	0.949
		✗	0.202	3.657	12.928	0.338	0.765	0.879	0.942
	– LSD	✓	0.212	4.101	13.809	0.319	0.748	0.852	0.936
		✗	0.224	4.231	14.771	0.335	0.726	0.838	0.923
	All	✓	0.160	2.610	10.814	0.258	0.811	0.924	0.961
		✗	0.161	2.633	11.034	0.272	0.813	0.915	0.956
nuScenes	– TA	✓	0.250	3.912	7.258	0.330	0.741	0.881	0.931
		✗	0.266	4.161	7.494	0.341	0.738	0.879	0.928
	– PD	✓	0.255	3.812	7.468	0.342	0.727	0.865	0.919
		✗	0.266	4.239	7.629	0.354	0.712	0.853	0.907
	– W	✓	0.266	4.323	7.925	0.375	0.708	0.846	0.904
		✗	0.281	5.779	8.206	0.418	0.688	0.825	0.883
	– LSD	✓	0.278	4.411	8.328	0.409	0.671	0.827	0.888
		✗	0.303	6.462	8.858	0.421	0.655	0.806	0.861
	All	✓	0.236	3.566	7.054	0.311	0.747	0.891	0.941
		✗	0.255	4.730	7.205	0.326	0.746	0.885	0.935

Table 7: **ZeroDepth outdoor depth estimation results using different training datasets.** *All* refers to the use of all 4 considered datasets, and each additional entry indicates the removal of a specific dataset: TA for *TartanAir*, PD for *Parallel Domain*, W for *Waymo*, and LSD for *Large-Scale Driving*. We observe a consistent decrease in performance when fewer training datasets are considered, and this decrease is similar between metric and median-scaled predictions.

modal scale consistency and awareness for monocular self-supervised depth estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE (in press), 2021. 6

[9] Xingshuai Dong, Matthew A. Garratt, Sreenatha G. Anavatti, and Hussein A. Abbass. Towards real-time monocular depth estimation for robotics: A survey, 2021. 1

[10] 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*, pages 10786–10796, 2021. 1, 2, 5, 7

[11] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction using a multi-scale deep network. *arXiv:1406.2283*, 2014. 2, 5

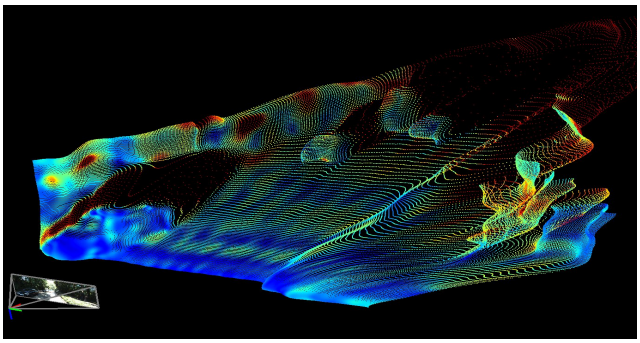
[12] Jose M. Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. CAM-Convs: Camera-Aware Multi-Scale Convolutions for Single-View Depth. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[13] Jiading Fang, Igor Vasiljevic, Vitor Guizilini, Rares Ambrus, Greg Shakhnarovich, Adrien Gaidon, and Matthew Walter. Self-supervised camera self-calibration from video. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022. 2

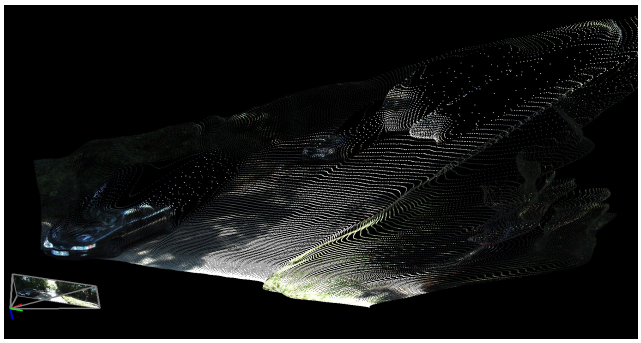
[14] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 1

[15] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 5

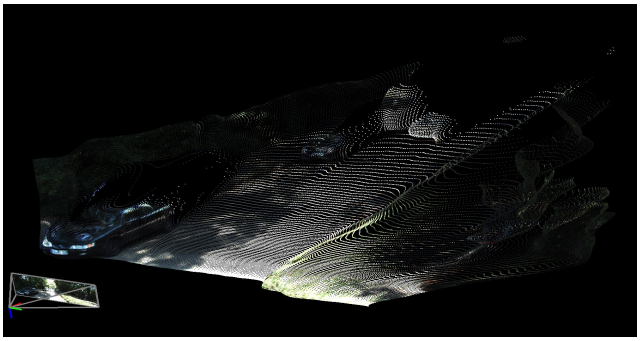
[16] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 5, 6



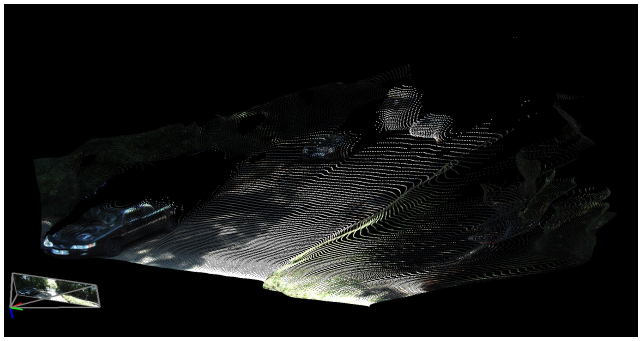
(a) Standard deviation



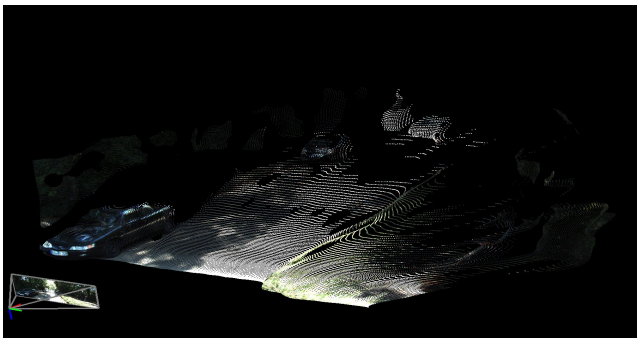
(b) 100% (RMSE 4.044)



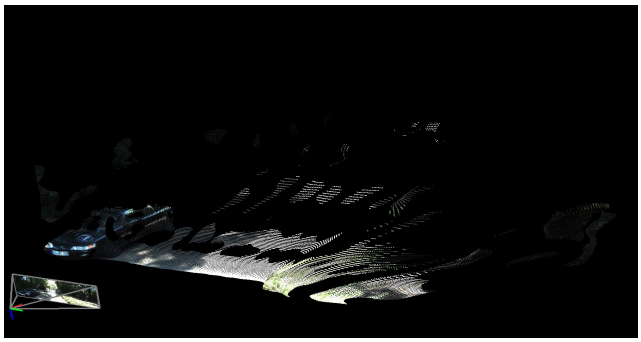
(c) 90% (RMSE 2.859)



(d) 75% (RMSE 2.132)

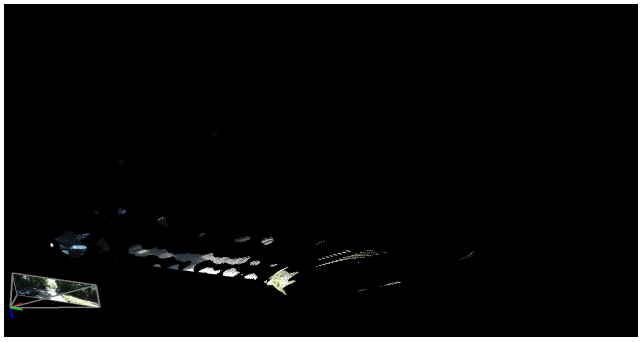


(e) 50% (RMSE 1.489)



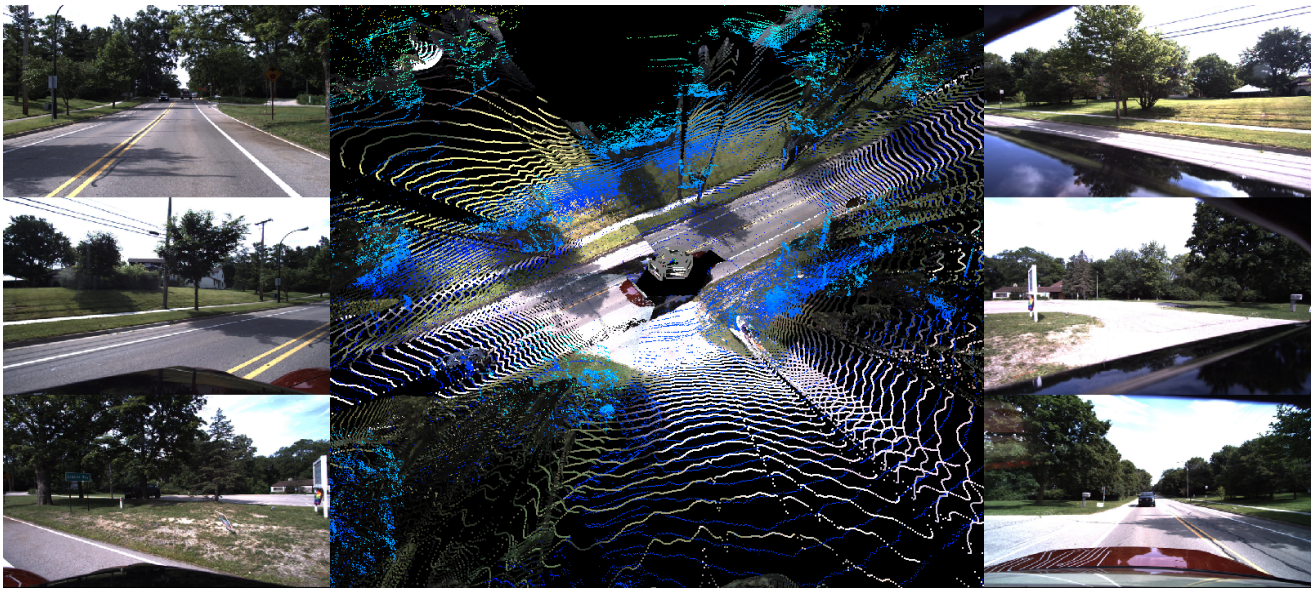
(f) 25% (RMSE 1.174)

(g) 10% (RMSE 0.723)

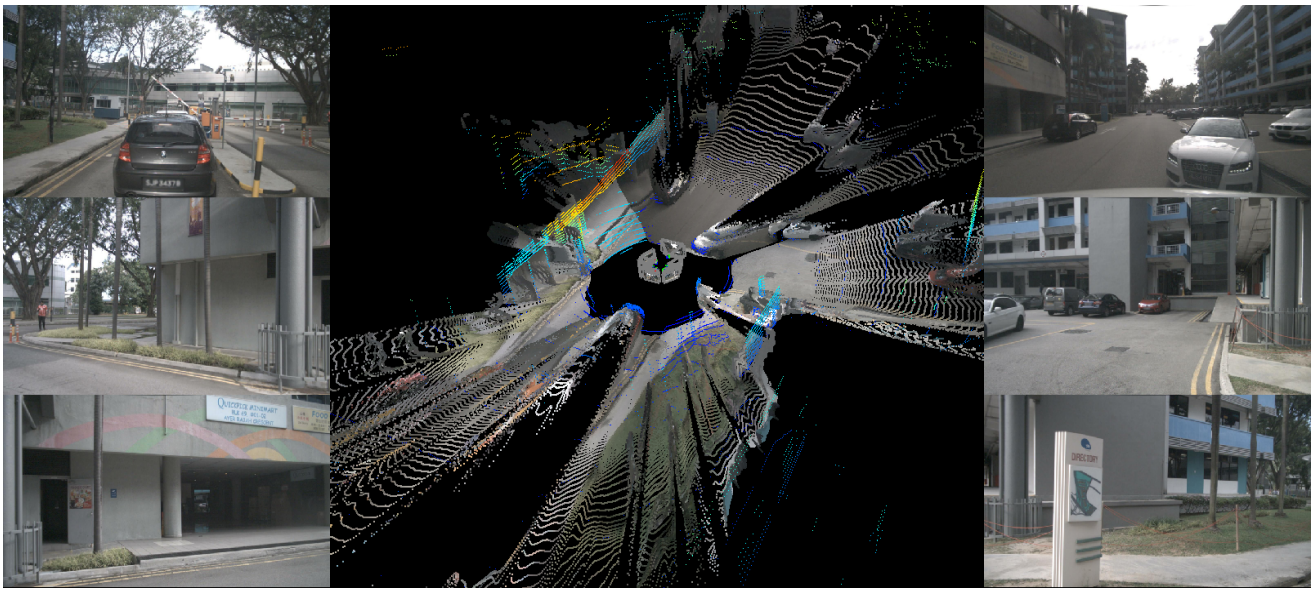


(h) 5% (RMSE 0.529)

Figure 9: **ZeroDepth pointcloud filtering based on variational uncertainty.** In (a) we show the predicted monocular pointcloud colored based on the standard deviation calculated from 10 samples. Afterwards, we show the same pointcloud filtered according to standard deviation (lowest to highest), and also report the corresponding RMSE from the filtered depth map. Even with minimal filtering (e.g., 10%) we already observe significant improvements (30%) in accuracy, mostly by removing areas with "bleeding" artifacts due to object discontinuities.



(a) DDAD



(b) nuScenes

Figure 10: **ZeroDepth full surround metric pointclouds**, obtained by overlaying predicted monocular pointclouds from the six available cameras on the (a) *DDAD* and (b) *nuScenes* datasets. LiDAR pointclouds are shown as height maps for comparison purposes only. No post-processing, scaling, or alignment of any kind was performed. More examples are shown in our supplementary video.

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2

[18] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 2

[19] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. In *ICCV*, 2019. 1, 2, 4, 6, 7, 9, 10

[20] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *CVPR*, 2019. 2

[21] Vitor Guizilini, Rares Ambrus, Wolfram Burgard, and Adrien Gaidon. Sparse auxiliary networks for unified monocular depth prediction and completion. In *IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 8

[22] Vitor Guizilini, Rares Ambrus, Dian Chen, Sergey Zakharov, and Adrien Gaidon. Multi-frame self-supervised depth with transformers. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[23] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *CVPR*, 2020. 1, 2, 4, 5, 6, 11

[24] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *ICLR*, 2020. 2, 6

[25] Vitor Guizilini, Kuan-Hui Lee, Rares Ambrus, and Adrien Gaidon. Learning optical flow, depth, and scene flow without real-world labels. *IEEE Robotics and Automation Letters*, 2022. 2, 5

[26] Vitor Guizilini, Jie Li, Rares Ambrus, and Adrien Gaidon. Geometric unsupervised domain adaptation for semantic segmentation. In *ICCV*, 2021. 1, 4, 5, 6

[27] Vitor Guizilini, Igor Vasiljevic, Rares Ambrus, Greg Shakhnarovich, and Adrien Gaidon. Full surround monodepth from multiple cameras. *arXiv:2104.00152*, 2021. 1, 2, 6, 7, 10

[28] Vitor Guizilini, Igor Vasiljevic, Jiading Fang, Rares Ambrus, Greg Shakhnarovich, Matthew Walter, and Adrien Gaidon. Depth field networks for generalizable multi-view scene representation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2022. 3, 4

[29] Akhil Gurram, Ahmet Faruk Tuna, Fengyi Shen, Onay Urfalioglu, and Antonio M López. Monocular depth estimation through virtual-world supervision and real-world sfm self-supervision. *arXiv:2103.12209*, 2021. 1, 6

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3, 9

[31] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 9

[32] Muhamamd Ishfaq Hussain, Muhammad Aasim Rafique, and Moongu Jeon. Rvmde: Radar validated monocular depth estimation for robotics, 2021. 1

[33] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver IO: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021. 2, 3, 10

[34] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5885–5894, October 2021. 9

[35] P. Ji, R. Li, B. Bhanu, and Y. Xu. Monoindoor: Towards good practice of self-supervised monocular depth estimation for indoor environments. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 7

[36] Jiaqi Zou Ke Mei, Chuang Zhu and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *European Conference on Computer Vision (ECCV)*, 2020. 1

[37] Jung Hee Kim, Junhwa Hur, Tien Phuoc Nguyen, and Seong-Gyun Jeong. Self-supervised surround-view depth estimation with volumetric feature fusion. In *Advances in Neural Information Processing Systems*, 2022. 2, 6, 7

[38] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 1951. 3

[39] 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. *arXiv preprint arXiv:1907.10326*, 2019. 1, 4

[40] 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. *arXiv:1907.10326*, 2019. 1, 8

[41] Kuan-Hui Lee, German Ros, Jie Li, and Adrien Gaidon. Spigan: Privileged adversarial learning from simulation. In *corl*, 2019. 1

[42] Sihaeng Lee, Janghyeon Lee, Byungju Kim, Eojindl Yi, and Junmo Kim. Patch-wise attention network for monocular depth estimation. In *In Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. 8

[43] Boying Li, Yuan Huang, Zeyu Liu, Danping Zou, and Wenxian Yu. Structdepth: Leveraging the structural regularities for self-supervised indoor depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 7

[44] Runze Li, Pan Ji, Yi Xu, and Bir Bhanu. Monoindoor++:towards better practice of self-supervised monocular depth estimation for indoor environments. *ArXiv*, abs/2207.08951, 2022. 7

[45] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv preprint arXiv:2204.00987*, 2022. 8

[46] Ce Liu, Suryansh Kumar, Shuhang Gu, Radu Timofte, and Luc Van Gool. Va-depthnet: A variational approach to single image depth prediction, 2023. 8

[47] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 9

[48] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 2, 5, 7

[49] 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 32*. 2019. 9

[50] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *arXiv:2103.13413*, 2021. 1, 4, 7, 9, 10

[51] 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*, 2022. 1, 2

[52] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *ECCV*, 2020. 6

[53] 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*, pages 2446–2454, 2020. 5

[54] Kunal Swami, Amrit Muduli, Uttam Gurram, and Pankaj Bajpai. Do what you can, with what you have: Scale-aware and high quality monocular depth estimation without real world labels. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022. 6

[55] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity invariant cnns. In *3DV*, 2017. 8

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

[57] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Mathieu Cord, and Patrick Pérez. Dada: Depth-aware domain adaptation in semantic segmentation. In *ICCV*, 2019. 1

[58] Brandon Wagstaff and Jonathan Kelly. Self-supervised scale recovery for monocular depth and egomotion estimation. In *IROS*, 2021. 2

[59] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *IROS*, 2020. 5

[60] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Yongming Rao, Guan Huang, Jiwen Lu, and Jie Zhou. Surrounddepth: Entangling surrounding views for self-supervised multi-camera depth estimation. *arXiv preprint arXiv:2204.03636*, 2022. 1, 2, 6, 7, 10

[61] Cho-Ying Wu, Jialiang Wang, Michael Hall, Ulrich Neumann, and Shuochen Su. Toward practical monocular indoor depth estimation. In *CVPR*, 2022. 1, 2, 7

[62] Wang Yifan, Carl Doersch, Relja Arandjelović, João Carreira, and Andrew Zisserman. Input-level inductive biases for 3D reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2022. 3

[63] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *ICCV*, 2019. 1

[64] Zhedong Zheng and Yi Yang. Unsupervised scene adaptation with memory regularization in vivo. In *IJCAI*, 2020. 1

[65] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 2