GRIN: Zero-Shot Metric Depth with Pixel-Level Diffusion

Vitor Guizilini Pavel Tokmakov Achal Dave Rares Ambrus

Toyota Research Institute (TRI)

{first.lastname}@tri.global

Abstract

3D reconstruction from a single image is a long-standing problem in computer vision. Learning-based methods address its inherent scale ambiguity by leveraging increasingly large labeled and unlabeled datasets, to produce geometric priors capable of generating accurate predictions across domains. As a result, state of the art approaches show impressive performance in zero-shot relative and metric depth estimation. Recently, diffusion models have exhibited remarkable scalability and generalizable properties in their learned representations. However, because these models repurpose tools originally designed for image generation, they can only operate on dense ground-truth, which is not available for most depth labels, especially in real-world settings. In this paper we present GRIN, an efficient diffusion model designed to ingest sparse unstructured training data. We use image features with 3D geometric positional encodings to condition the diffusion process both globally and locally, generating depth predictions at a pixel-level. With comprehensive experiments across eight indoor and outdoor datasets, we show that GRIN establishes a new state of the art in zero-shot metric monocular depth estimation even when trained from scratch.

1. Introduction

Depth estimation is a fundamental problem in computer vision and a core component of many practical applications, including augmented reality [16], medical imaging [43] mobile robotics [10, 34], and autonomous driving [15, 23, 41]. In reality, most of these applications benefit from *metric* depth estimates, that capture the true physical shape of the observed environment (i.e., in meters) and enable scale-aware 3D reconstruction. Although recovering metric depth is trivial in the multi-view calibrated setting [30], only recently it started to be explored in the monocular context. In this ill-posed setting, models must learn priors from training data in order to reason over the scale ambiguity and generate accurate predictions. The challenges with this approach are two-fold: (i) the choice of priors themselves, that should be expressive enough to generalize across diverse domains; and (ii) the choice of network architecture, that should be capable of detecting and learning these priors from large-scale diverse training data.

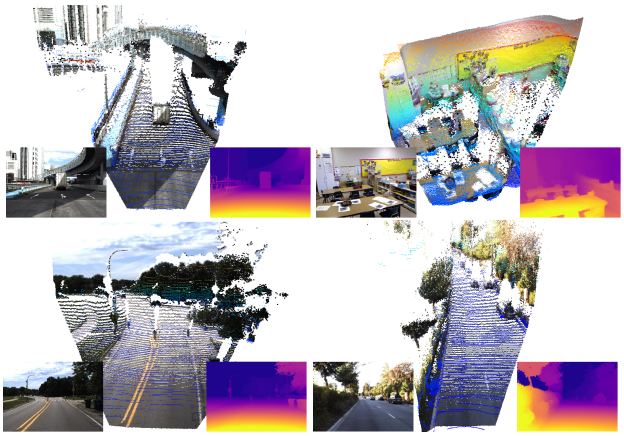


Figure 1. **GRIN sets a new state of the art** in zero-shot metric monocular depth estimation, via efficient pixel-level diffusion and the proper handling of sparse training data. For comparison, we overlay ground-truth metric data with predicted pointclouds.

In this work, we use input-level geometric embeddings from calibrated cameras [27] to learn physically-grounded priors capable of the zero-shot transfer of metric depth across datasets. In order to fully leverage these geometric priors we turn to diffusion models [31], due to their scalability to large-scale diverse datasets and strong regression performance in generative tasks, as well as improved generalization. This choice is becoming increasingly popular, with several published papers [36, 38, 54, 55] using diffusion models for monocular depth estimation. However, all these methods repurpose currently available diffusion frameworks, that are based on the U-Net architecture [52], and require compromises and *ad-hoc* solutions to adapt to this new setting. Broadly speaking, these compromises are two-fold: (i) the use of latent auto-encoders, that now must be trained on much smaller and less diverse datasets; and (ii) the need for dense ground-truth, which is not available for most real-world datasets.

To mitigate these limitations, we instead propose to use a more flexible diffusion architecture that is efficient enough to operate at a pixel-level, and can directly ingest sparse unstructured training data. In particular, we build on RIN (Recurrent Interface Networks) [35], a novel diffusion architecture that *decouples its core computation from input dimensionality*, making it much more efficient than traditional U-Net models; and that is *domain-agnostic*, thus not restricted to dense grid-like inputs. We propose several key modifications to this original framework to apply it to the task of depth estimation, including the use of 3D geometric positional encodings to bridge the geometric domain gap across datasets, a combination of local and global diffusion conditioning with dropout and random masking, and a log-space depth parameterization designed to improve performance in widely different ranges. As a result, our proposed Geometric RIN (GRIN) framework establishes a new state of the art in zero-shot metric monocular depth estimation. In summary, our contributions are as follows:

- We introduce **GRIN**, a novel diffusion-based monocular depth estimation framework designed to (i) ingest **sparse training data**, enabling the use of larger and more diverse datasets; and (ii) operate on **pixel-space**, eliminating the need for dedicated auto-encoders.
- We propose a combination of **local and global conditioning**, in the form of image features with 3D geometric positional encodings, to enable training and evaluation on datasets with **diverse camera geometries**.
- With extensive experiments across 8 different indoor and outdoor datasets, GRIN establishes a new **state of the art in zero-shot metric depth estimation**.

2. Related Work

2.1. Monocular Depth Estimation

Monocular depth estimation is the task of regressing per-pixel range from a single image. Early learning-based approaches were fully supervised [13, 14], requiring datasets with annotations from additional range sensors such as IR [46] or LiDAR [18]. Although ground-breaking at the time, these methods lacked scalability, due to the need for dedicated hardware, as well as high sparsity and noise levels in the collected labels. The seminal work of [82] introduced the concept of self-supervision to monocular depth estimation, eliminating the need for explicit supervision in favor of a multi-view photometric objective. This approach is highly scalable, since it only requires overlapping images, and further developments [19, 20, 22–24, 57, 70] have led to accuracy comparable with supervised approaches. However, self-supervision also has its drawbacks, due to inherent limitations in the multi-view photometric objective itself, and most notably the inability to generate metric estimates due to scale ambiguity [23, 26, 37].

Recently, a sharp increase in publicly available datasets [4, 8, 12, 23, 64, 69] gave rise to a third approach: large-scale supervised pre-training to generate a rich visual representation that can be transferred to new domains with minimal to no fine-tuning [12, 50, 77]. In this setting, the challenge becomes how to design such a visual representation, so it can learn robust and transferable priors [27] capable of bridging the *appearance* and *geometric* domain gaps. This includes both architectures [27, 55, 76] as well as the application; i.e. relative [12, 38, 50] or metric [27, 55, 76] depth, focusing on a different set of learned priors.

2.2. Zero-Shot Metric Depth Estimation

Several works have explored ways to generate metric predictions without explicit supervision in the target domain. Self-supervised methods [14, 81, 82] require the indirect injection of metric information, obtained from different sources such as velocity measurements [23], camera height [68], cross-camera extrinsics [26, 39, 71], or left-right stereo consistency [73]. Recently, a few works have explored the zero-shot transfer of metric predictions across datasets. ZoeDepth [2] fine-tunes a scale-invariant model in a combination of indoor and outdoor datasets, learning domain-specific decoders with adaptive ranges. Metric3D[76] proposes a canonical camera space transformation module, that abstracts away scale ambiguity during training in favor of a post-processing scale alignment step. ZeroDepth [27] takes a different approach and, instead of abstracting away camera intrinsics, uses it as input-level geometric embeddings to learn 3D scale priors over objects and scenes. DMD [55] uses a similar field-of-view conditioning approach, in combination with synthetic augmentation to increase camera diversity. UniDepth [49] chooses instead to directly predict 3D points, relying on a pseudo-spherical output space to also estimate camera parameters.

2.3. Diffusion Models for Depth Estimation

Denoising Diffusion Probabilistic Models (DDPM) [31] are a class of generative models that have become very popular recently. Their aim is to reverse a diffusion process, generating samples from a target distribution by learning how to iteratively denoise a random Gaussian distribution. Although originally proposed for image generation [9, 47, 62], several works have shown their effectiveness in other computer vision tasks, such as semantic segmentation [36], panoptic segmentation [6], optical flow [54], and monocular depth [11, 36, 38, 54–56, 74, 80].

Focusing on monocular depth estimation, DDP [36] operates in the latent space, using an input image as the conditioning signal. Similarly, DiffusionDepth [11] uses local and global multi-scale image features from a Swim Transformer [44]. DepthGen [56] proposes novel tools to handle noisy ground-truth, and DDVM [54] explores

self-supervised pre-training in combination with synthetic and real-world training data. A few concurrent works also look into zero-shot diffusion-based depth estimation. Marigold [38] proposes to fine-tune pre-trained text-to-image generators with synthetic depth labels, focusing on affine-invariant predictions. DMD [55] uses field-of-view conditioning to handle scale ambiguity and enable the zero-shot transfer of metric depth.

Importantly, all these works rely on different techniques to address the sparsity of training data. These include: *infilling* (interpolating missing values) [11, 36, 54–56], *step-unrolling* (adding noise to the model output rather than the ground-truth) [54, 55], or *avoiding* sparse training data altogether [38, 74]. Conversely, GRIN does not require any of these techniques, since it was designed to ingest sparse training data without assuming any spatial structure. Furthermore, our efficient architecture enables pixel-level diffusion, thus eliminating the need for specialized auto-encoders and promoting sharper predictions.

3. Diffusion Preliminaries

We begin by providing a brief overview of diffusion models [5, 31, 58]. These methods were originally developed for image generation, operating via a series of learned state transitions from a noise tensor $\mathbf{N}_1$ to an image $\mathbf{I}_0$ from the data distribution. To learn this transition f, a forward function is first defined as:

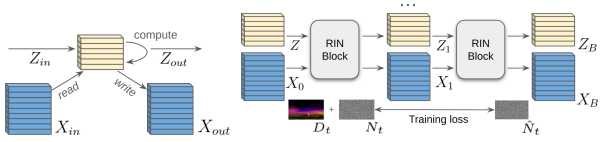
$$\mathbf{I}_t = \sqrt{\gamma(t)}\mathbf{I}_0 + \sqrt{1-\gamma(t)}\mathbf{N}_1, \quad (1)$$

where $\mathbf{N}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $t \sim \mathcal{U}(0, 1)$ and $\gamma(t)$ is a monotonically decreasing function. A neural network is learned to predict $\mathbf{N}_t$ from $\mathbf{I}_t$ in a given transition step t via:

$$\tilde{\mathbf{N}}_t = f(\mathbf{I}_t, t) = f(\sqrt{\gamma(t)}\mathbf{I}_0 + \sqrt{1-\gamma(t)}\mathbf{N}_1, t) \quad (2)$$

and used to sample an image via a sequence of state transitions from $\mathbf{I}_1 = \mathbf{N}_1$ to $\mathbf{I}_0$ via small steps $\mathbf{I}_1 \rightarrow \mathbf{I}_{1-\Delta} \rightarrow ... \rightarrow \mathbf{I}_0$ [31, 59]. In practice, the diffusion process is often conditioned by an additional variable y, such as a class label [9], language caption [51], or camera parameters [42], to control the generated samples.

A central question when designing diffusion approaches is the choice of architecture for the transition function f. Mainstream methods [9, 32, 51] have used the U-Net CNN architecture [52] due to its simplicity and ability to preserve input resolution. However, this approach quickly becomes computationally prohibitive for high-resolution images. Because of that, most methods train f not in the RGB pixel space, but in a lower-resolution latent space produced by an auto-encoder [65]. Although more efficient, this approach also has its drawbacks, namely the loss of fine-grained details due to latent compression, and the assumption that inputs will be represented on a dense 2D grid, which is natural for images, but not for sparse data such as depth maps.



(a) RIN block. (b) RIN model.

Figure 2. **Recurrent Interface Networks (RIN) architecture.** (a) Latent tokens $\mathbf{Z}_{in}$ read from input tokens $\mathbf{X}_{in}$, are processed via a series of self-attention layers, and written back to output tokens $\mathbf{X}_{out}$. (b) A RIN model consists of B blocks, each receiving latent $\mathbf{Z}_b$ and input $\mathbf{X}_b$ tokens from the previous block and returning updated $\mathbf{Z}_{b+1}$ and $\mathbf{X}_{b+1}$.

Recurrent Interface Networks (RIN). To circumvent these limitations, we instead adopt RIN [35], a recently introduced transformer-based architecture, shown in Figure 2. The key idea behind RIN is the separation of computation into input tokens $\mathbf{X} \in \mathbb{R}^{N \times D}$ and latent tokens $\mathbf{Z} \in \mathbb{R}^{M \times D}$, where the former is obtained by tokenizing input data (and hence N is dependent on input size), but M is a fixed dimension. The computation is then performed via a sequence of attention operations. First, the latents $\mathbf{Z}$ attend to inputs $\mathbf{X}$ (*read* operation), followed by several self-attention operations on $\mathbf{Z}$ (*compute*) and the final *write* from latents to inputs. This forms a single RIN block (Figure 2a), and stacking multiple blocks enables the construction of deeper models (Figure 2b, please refer to [35] for further details).

The fact that the computation cost of RIN is independent of input size enables us to learn the transition function directly in pixel space. Moreover, the tokenization step removes the requirement for inputs to be represented on a dense grid. Capitalizing on these benefits, in the next section we introduce our approach for zero shot metric depth estimation with pixel-level diffusion.

4. Geometric RIN

We propose the GRIN (Geometric RIN) architecture, as shown in Figure 3. GRIN takes as input a noisy single-channel depth map $\mathbf{D} \in \mathbb{R}^{H \times W}$, containing pixel-wise d_{jk} distances to the camera ranging between d_s and d_f, for $j \in [0, H]$ and $k \in [0, W]$ and outputs the estimated noise matrix $\mathbf{N}_t$. Importantly, the depth values are *metric*, representing physical distances, and we make the design choice of working with *euclidean* depth, representing distance along the viewing ray $\mathbf{r}_{jk}$, rather than the more traditional *z-depth* parameterization. Moreover, $\mathbf{D}$ is assumed to be *sparse*, meaning that specific d_{jk} can potentially be missing. We describe how we address the sparsity challenge in Section 4.1. The denoising process is conditioned on an RGB image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ and corresponding camera intrinsics $\mathbf{K}$. The design of these conditioning vectors is a key component of GRIN, and is described in details in Sections 4.2 and 4.3.

4.1. Sparse Unstructured Training

Differently from traditional U-Net architectures, RIN does not assume any spatial structure in its input tokens $\mathbf{X}$. This is necessary to enable training from sparse unstructured data, where there is no explicit concept of neighborhood. In GRIN, spatial structure is defined by geometric embeddings used as conditioning, and once incorporated each token is treated independently, which enables processing only parts of the input with available ground truth.

Concretely, during training, we assume ground-truth in the form of a 2D grid $\mathbf{D} \in \mathbb{R}^{H \times W}$ with $N < HW$ valid pixels. Each valid depth value d_{jk} is paired with the corresponding RGB pixel value $\mathbf{p}_{jk} = (u, v)_{jk}$ and geometric embedding $\mathbf{g}_{ijk}$ for conditioning (see Section 4.2 for details). Note, however, that in the case of very sparse labels ($N << HW$), this could result in few remaining pixels, limiting the amount of information about the scene context. Moreover, some areas will never produce valid depth labels for supervision (e.g., the sky). To address these limitations we propose a combination of local and global conditioning which promote training with unstructured sparse data while still maintaining dense scene-level information.

4.2. GRIN Embeddings

We use two input modalities to condition depth predictions during the GRIN diffusion process: images and camera geometry. Although image-level conditioning has already been widely used in diffusion models, enabling tasks such as image-to-image translation [53, 79] and even in-domain [36, 54] or affine-invariant [38, 74] depth estimation, the use of camera information has only recently started to be explored [42, 55]. GRIN differs from these methods in the sense that camera information is used to condition predictions at a pixel-level, rather than globally (i.e., camera extrinsics in [42] and focal length in [55]). Below we describe each one of these embeddings in detail.

Image Embeddings are generated using an encoder $\mathcal{F}_\theta$, with learnable parameters θ, to process an input image $\mathbf{I}$ such that $\mathbf{F} = \mathcal{F}_\theta(I)$. Following RIN [35], we use a single convolutional layer $\mathcal{F}^{loc}_\theta$, with kernel size $K \times K$ and C_l output channel dimensions, to directly tokenize $\mathbf{I}$. This results in a flattened $\mathbf{F}^{loc} \in \mathbb{R}^{HW \times C_l}$ feature map containing patch-wise visual information $\mathbf{f}_{jk}$ for each pixel $\mathbf{p}_{jk} = (u, v)_{jk}$ within $\mathbf{I}$.

Geometric Embeddings are generated using information from the camera used to obtain $\mathbf{I}$, in the form of a 3×3 intrinsic $\mathbf{K}$ matrix (assumed to be pinhole for simplicity, although any geometric model can be readily used). Each pixel $\mathbf{p}_{jk}$ from image $\mathbf{I}$ is parameterized in terms of its viewing ray $\mathbf{r}_{jk} = \mathbf{K}^{-1} [u_{jk}, v_{jk}, 1]^T$, with the camera center assumed to be at the origin $\mathbf{t}_{jk} = [0, 0, 0]^T$. To increase expressiveness, we follow the standard approach [27, 28, 45] of Fourier-encoding these values. Assuming N_o encoding frequencies for camera centers and N_r for viewing rays, the resulting geometric embeddings are of dimensionality $D = 2\big(3(N_o + 1) + 3(N_r + 1)\big) = 6\,(N_o + N_r + 2)$. The resulting embeddings $\mathbf{g}_{ijk} = \mathcal{G}(\mathbf{t}_i, \mathbf{r}_{ijk}) = \mathcal{E}(\mathbf{t}_i) \oplus \mathcal{E}(\mathbf{r}_{ijk})$, where $\oplus$ denotes concatenation, are used to imbue visual information with geometric awareness, resulting in features capable of reasoning over 3D properties such as physical shape and scale. As shown in previous works, this is a key enabler of capabilities such as implicit learning of multi-view geometry [28, 29, 75] and zero-shot transfer of metric depth across datasets with diverse cameras [27].

Depth Embeddings are generated from ground-truth labels during training, and estimated as predictions during inference. To enable learning from sparse unstructured data, GRIN operates at a pixel-level, and therefore does not require latent auto-encoders or tokenizers. However, in agreement with [55], we have independently verified that a log-scale parameterization leads to improved results when dealing with large range intervals. Specifically, our projection and unprojection functions mapping d_{jk} to and from log-space $\hat{d}_{jk}$ are defined as:

$$\hat{d}_{jk} = \log_b \left((b-1) \frac{d_{jk} - d_s}{d_f - d_s} + 1 \right) \tag{3}$$

$$d_{jk} = \frac{b^{\hat{d}_{jk}} - 1}{b - 1} (d_f - d_s) + d_s \tag{4}$$

where b is the logarithm base, that determines how distances will be compressed at different ranges. Our goal is to make shorter ranges *more robust to residual noise from the diffusion process*, without compromising performance at longer ranges, where this residual noise is less impactful. In our ablation analysis (Section 5.6) we evaluate different b values, as well as a linear parameterization.

4.3. GRIN Conditioning

Local Conditioning. To condition the denoising process with the image and geometric embeddings defined above, we simply concatenate them to the corresponding depth embeddings in the token dimension. As a result, in GRIN we make a simple yet crucial design choice and, instead of traditional positional encodings [67], that describe only the 2D location $(u, v)_{jk}$ of each pixel $\mathbf{p}_{jk}$ within $\mathbf{I}$, we use geometric embeddings $\mathbf{g}_{jk}$ to describe each pixel in a 3D reference frame. This choice guides the denoising process not only towards localized predictions within the image, but also promotes disambiguation between camera geometries (e.g., focal length, resolution, or distortion). For a 1-dimensional prediction $\mathbf{d}_{jk} \in \mathbb{R}$, the conditioned vector is defined as $\hat{\mathbf{d}}_{jk} = \mathbf{d}_{jk} \oplus \mathbf{f}^{loc}_{jk} \oplus \mathbf{g}_{jk}$, projected onto a V-dimensional vector $\mathbf{v}_{jk}$ using a linear layer $\mathcal{P}^{loc}_{1+C_l+D \rightarrow V}$. The collection of conditioned vectors for all HW predictions to be estimated during the denoising process is given by $\mathbf{V}^{loc} \in \mathbb{R}^{HW \times V}$.

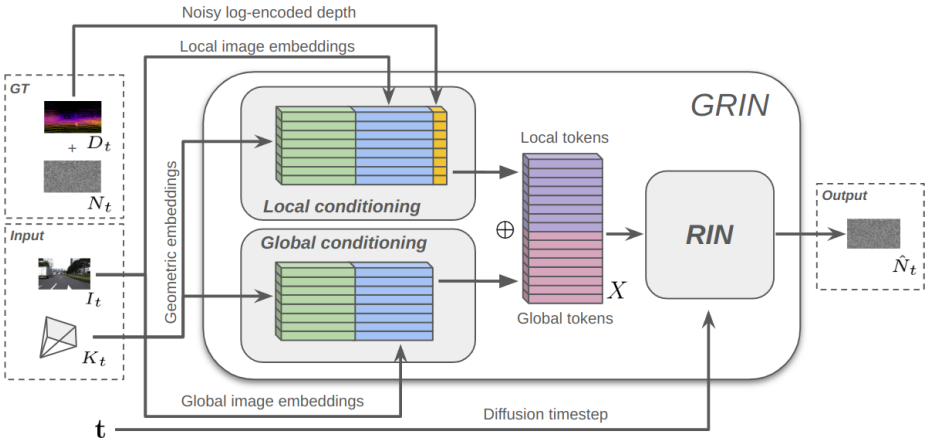


Figure 3. **Diagram of GRIN for monocular depth estimation.** An input image $\mathbf{I}$ with intrinsics $\mathbf{K}$ is used to condition the diffusion process both *locally*, by augmenting each pixel to be predicted with geometrically aware visual features; and *globally*, by introducing additional scene-level information decoupled from the pixels to be predicted. The resulting tokens are concatenated and attended with the RIN latent space, generating noise predictions for a particular diffusion timestep.

Global Conditioning. Global image embeddings are generated using a convolutional encoder $\mathcal{F}_\theta^{glob}$, resulting in multi-scale feature maps $\mathbf{F}^{glob} = [\mathbf{F}^0, \mathbf{F}^1, ..., \mathbf{F}^S]$ at S increasingly lower resolutions. Lower-resolution feature maps are upsampled, concatenated and flattened to generate $\hat{\mathbf{F}}^{glob} \in \mathbb{R}^{\frac{HW}{d^2} \times C_g}$, where d is the downsampling factor of the highest encoded resolution and C_g is the concatenated channel-wise dimension. These embeddings contain scene-level multi-resolution visual information that is not tied to any specific pixel-level prediction, but rather used to promote global consistency during the denoising process. To promote spatial structure, we use a combination of image $\hat{\mathbf{F}}^{glob}$ and geometric $\mathbf{G}$ embeddings, the latter generated from a camera resized to match the former's resolution. Similarly to local conditioning, concatenated embeddings are projected onto V-dimensional vectors using a linear layer $\mathcal{P}^{glob}_{C_g+D \to V}$. The collection of M vectors used to globally condition the denoising process is given by $\mathbf{V}^{glob} \in \mathbb{R}^{M \times V}$, and concatenated with $\mathbf{V}^{loc}$ to produce input tokens $\mathbf{X} = \mathbf{V}^{loc} \oplus \mathbf{V}^{glob} \in \mathbb{R}^{(N+M) \times V}$.

4.4. Training Procedure

At training time we discard pixels with missing depth information (i.e., $d_{jk} = 0$), resulting in a $\hat{\mathbf{V}}^{loc}$ matrix with varying length N. To improve iteration speed, we also randomly discard a percentage of valid local vectors, thus only supervising on a subset of L pixels, which leads to faster cross-attention with the GRIN latent tokens. This is akin to training on image crops, taken to the extreme by supervising instead on a random subset of pixels. A similar process is applied to the global vector matrix $\hat{\mathbf{V}}^{glob}$ (i.e., only using a subset G of global vectors), which we have empirically observed (Section 5.6) that not only leads to faster iteration speeds but also improves performance. This is akin to dropout [63], which is known to improve generalization. The resulting input tokens are of dimensionality $\hat{\mathbf{X}} \in \mathbb{R}^{(L+G) \times V}$. Our training objective is the L2 loss, calculated in this log-depth scale such that $\mathcal{L}(t) = \left(\mathbf{N}_t \gamma(t) - \tilde{\mathbf{N}}_t\right)^2$, where $\mathbf{N}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the injected noise at timestep $t \sim \mathcal{U}(0, 1)$ and $\tilde{\mathbf{N}}_t$ is the GRIN predicted noise at that timestep. For more information, we refer the reader to the supplementary material.

4.5. Inference Procedure

At inference time we use the full $\mathbf{V}^{loc}$ and $\mathbf{V}^{glob}$ matrices, to maximize the amount of available information, although this is not strictly necessary. In the supplementary material we ablate the partial use of global vectors during inference, and show that targeted depth estimation can be done by only considering a subset of local vectors (i.e., to estimate depth only on image crops, such as 2D bounding boxes). A random noise matrix $\mathbf{N}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \in \mathbb{R}^{HW}$ is then sampled, and conditioned both locally and globally to produce input tokens $\mathbf{X} \in \mathbb{R}^{(HW+M) \times V}$ for GRIN. During the denoising process, at each timestep t a noise prediction $\hat{\mathbf{N}}_t$ is used to guide the generation of depth values for each input patch. After T iterations, the resulting $\tilde{\mathbf{V}}^{loc}$ local vectors are extracted from $\tilde{\mathbf{X}}$ and projected onto a 1-channel vector containing log-scaled depth predictions, using a linear layer $\mathcal{P}^{dec}_{V \to 1}$. These predictions are then converted to linear depth estimates using Equation 4.

5. Experiments

5.1. Training Datasets

We trained GRIN using a diverse combination of indoor and outdoor datasets from both real-world and syn-

	AbsRel↓	RMSE↓	$\delta < 1.25$↑	AbsRel↓	RMSE↓	$\delta < 1.25$↑	AbsRel↓	RMSE↓	$\delta < 1.25$↑	AbsRel↓	RMSE↓	$\delta < 1.25$↑
	KITTI [17]			*DDAD* [23]			*nuScenes* [4]			*VKITTI2* [3]		
AdaBins* [1]	0.058	2.360	0.964	0.147	7.550	0.766	0.445	10.658	0.471	0.133	6.248	0.803
NeWCRFs* [78]	0.052	2.129	0.974	0.119	6.183	0.874	0.400	12.139	0.512	0.117	5.691	0.829
ZeroDepth [27]	*0.064*	*2.987*	*0.958*	0.100	6.318	0.889	0.157	7.612	0.822	*0.099*	*4.209*	*0.905*
ZoeDepth† [2]	N/A	N/A	N/A	0.138	7.225	0.824	*0.198*	*8.245*	*0.809*	0.105	5.095	0.850
DMD [55]	N/A	N/A	N/A	0.108	5.365	0.907	N/A	N/A	N/A	0.092	4.387	0.890
Metric3D [76]	0.058	2.770	0.964	N/A	N/A	N/A	0.147	7.889	—	*0.089*	*4.201*	*0.904*
UniDepth [49]	0.047	**2.000**	0.980	*0.097*	*5.399*	*0.919*	*0.143*	*7.425*	*0.839*	*0.078*	*3.850*	*0.923*
GRIN	**0.046**	2.251	**0.983**	**0.093**	**5.307**	**0.922**	**0.138**	**7.217**	**0.857**	**0.074**	**3.501**	**0.937**
	NYUv2 [46]			*SunRGBD* [61]			*DIODE (indoor)* [66]			*DIODE (outdoor)* [66]		
AdaBins* [1]	0.103	0.364	0.903	0.159	0.476	0.771	0.443	1.963	0.174	0.865	10.350	0.158
NeWCRFs* [78]	0.095	0.334	0.922	0.151	0.424	0.798	0.404	1.867	0.187	0.854	9.228	0.176
ZeroDepth [27]	0.100	0.380	0.901	*0.121*	*0.347*	*0.864*	*0.309*	*1.779*	*0.377*	*0.714*	*7.880*	*0.219*
ZoeDepth† [2]	N/A	N/A	N/A	0.123	0.356	0.856	0.331	1.598	0.386	0.757	7.569	0.208
DMD [55]	N/A	N/A	N/A	0.109	0.306	0.914	0.291	1.292	0.380	0.553	8.943	0.187
Metric3D [76]	0.094	0.337	0.926	*0.104*	*0.319*	*0.919*	0.268	1.429	—	0.414	6.934	—
UniDepth [49]	0.063	0.232	**0.984**	*0.106*	*0.316*	*0.918*	*0.237*	*1.329*	*0.408*	*0.401*	*6.491*	*0.278*
GRIN	**0.058**	**0.209**	0.980	**0.098**	**0.301**	**0.927**	**0.221**	**1.128**	**0.439**	**0.393**	**6.011**	**0.303**

Table 1. **Zero-shot metric monocular depth estimation results** on various indoor and outdoor datasets. Numbers in *italics* indicate results obtained by evaluating specific methods on additional benchmarks using publicly available code and pre-trained models. UniDepth [49] was re-evaluated in most benchmarks because it does not report standard metrics in them (for a fair comparison, we used the *UniDepth-C* model, that also rely on input intrinsics and has the same ResNet backbone as ours). * indicates state-of-the-art methods trained and evaluated on the same dataset, for comparison. † indicates methods that do not require camera intrinsics. N/A indicate methods that cannot be evaluated zero-shot in a particular benchmark, because the benchmark dataset is used during training.

thetic sources. These include **Waymo** [64], with $990,340$ LiDAR-annotated images from 5 cameras, as a source of real-world driving data; **LyftL5** [33], with over $1,000$ hours of data collected by 20 self-driving cars, for a total $351,029$ LiDAR-annotated images from 7 cameras; **ArgoVerse2** [72], with $3,909,297$ LiDAR-annotated images from 7 cameras, for a total of $1,000$ sequences taken from the *Sensor* split; **Large-Scale Driving (LSD)** [27], with $1,057,920$ LiDAR-annotated images from 6 cameras, collected from multi-continental vehicles; **Parallel Domain (PD)** [24, 25], with $567,000$ images from 6 cameras containing procedurally generated photo-realistic renderings of urban driving scenes; **TartanAir** [69], with $613,274$ stereo images rendered from diverse synthetic scenes; **OmniData** [12], composed of a collection of synthetic datasets (Taskonomy, HM3D, Replica, and Replica-GSO), for a total of $14,340,580$ images from a wide range of environments and cameras; and **ScanNet** [8], with $547,991$ RGB-D samples collected from $1,413$ indoor scenes.

Note that most of these datasets contain sparse depth maps from LiDAR reprojection, which makes then unsuitable for traditional latent diffusion methods, but that can be directly ingested with our proposed pixel-level approach.

5.2. Implementation Details

Our models were implemented in PyTorch [48]. We used the LION optimizer [7], with batch size $b = 1024$, weight decay of $w_d = 10^{-2}$ (applied only to layer weights), $\beta_1 = 0.9$, $\beta_2 = 0.99$, and a warm-up scheduler [21] with linear increase for 10k steps followed by cosine decay. We use DDIM [60] with 1000 training and 10 evaluation timesteps, as well as EMA [40] with $\beta = 0.999$. Additional details are available in the supplementary material.

During training, input images (and intrinsics) are first resized to fit within a 640×512 resolution, and then randomly resized between $[0.5, 1.5]$ of this resolution, preserving aspect ratio. If the result is larger than 640×512 it is randomly cropped, otherwise it is padded, so it can be collated as part of a batch. The padded portions of each image are discarded during image embeddings calculation, and not used in the local and global conditioning stages. The same augmentation procedure is applied to ground-truth depth labels, albeit with different resizing parameters. We also apply horizontal flipping and color jittering as additional augmentations.

For efficiency purposes, training was conducted in two stages, for a total of 200k iterations steps. For the first 120k steps, a target resolution of 320×256 (half the original) was used. Moreover, for the first 40k steps only synthetic datasets were used, as a way to promote (a) sharper boundaries, due to the dense labels; and (b) reasoning over the full 200m range, with areas further away such as the sky being clipped to still serve as supervision. The remaining 80k steps used all training datasets, shuffled to ensure a similar ratio of indoor and outdoor samples per batch, as well as real-world and synthetic samples. The second stage used this same training strategy for an additional 80k steps, with

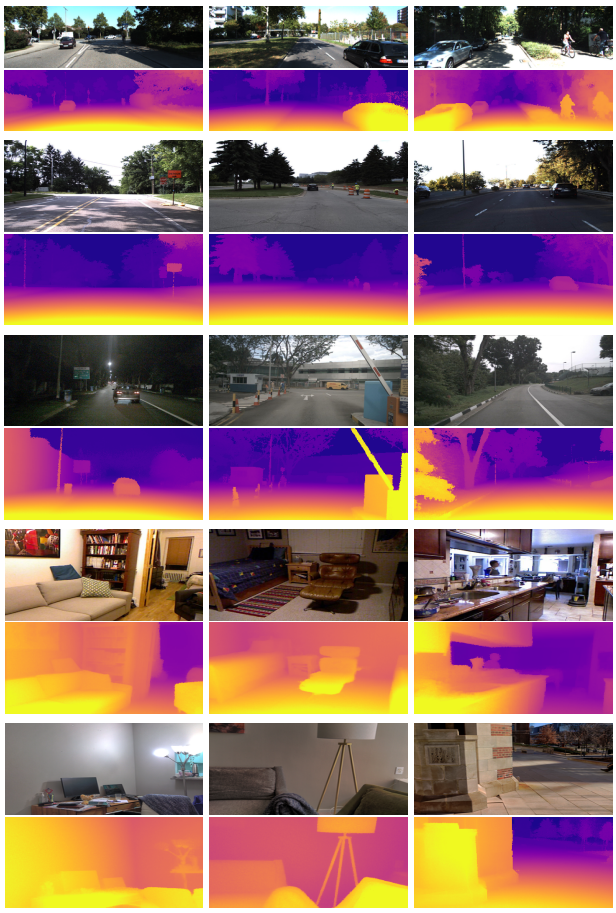


Figure 4. **Qualitative zero-shot metric depth estimation results using GRIN** on various indoor and outdoor datasets. The same model was used in all evaluations. For more examples, please refer to the supplementary material.

images at the target resolution and no additional changes. In total, training takes roughly 5 days with distributed data parallel (DDP) across 32 A100 GPUs, with mixed precision format. Inference for a 640×384 image can be done in 0.8 seconds on a single similar GPU (faster than Marigold).

5.3. Zero-Shot Metric Depth Estimation

We evaluated the zero-shot capabilities of GRIN on 8 standard indoor and outdoor monocular depth estimation benchmarks. These include **KITTI** [17], **VKITTI2** [3], **DDAD** [23], **nuScenes** [4], **DIODE** [66] (indoor and outdoor), **NYUv2** [46], and **SunRGBD** [61]. As baselines, we considered recently published state of the art methods [2, 27, 49, 55, 76] that also target zero-shot metric depth estimation. For a fair comparison, we used the standard evaluation protocol for each of these benchmarks, and when necessary re-evaluated models under the same conditions with official code and pre-trained checkpoints.

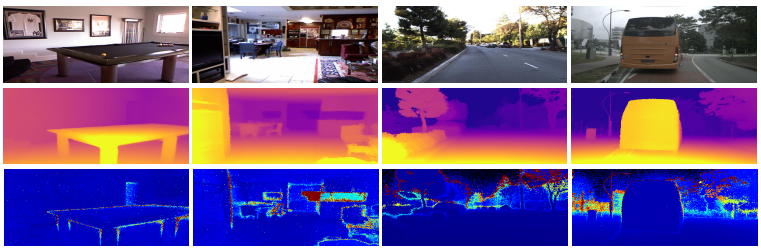
Quantitative results are reported in Table 1, showing that GRIN outperforms all considered methods and establishes a new state of the art in zero-shot metric monocular depth estimation. In particular, we outperform Metric3D [76], that proposes to overcome the geometric domain gap by projecting training data onto a canonical camera space. GRIN follows a different paradigm and instead exposes the network to this information, thus enabling the implicit learning of robust 3D-aware geometric priors that can be directly transferred across datasets. Interestingly, we also outperform ZeroDepth [27], that uses a similar approach to bridge the geometric domain gap. Similarly, we also outperform DMD [55], a diffusion-based approach that relies on field-of-view conditioning and synthetic data augmentation to increase camera diversity. We argue that our approach of directly ingesting sparse data is more scalable, since it enables supervised pre-training on much more diverse real-world datasets without relying on inaccurate preprocessing strategies to artificially generate dense ground-truth [11, 36, 38, 54–56, 74]. Lastly, we also outperform in almost all metrics (22 / 24) the very recent UniDepth [49], that directly predicts 3D points instead of depth maps, which enables the joint estimation of camera intrinsics. We believe GRIN could be modified to operate in a similar setting, which would potentially further improve performance, however this is left for future work. Qualitative examples of zero-shot GRIN predictions are shown in Figure 4.

Method	KITTI	NYUv2	DDAD	DIODE	ETH3D
Marigold	0.071	0.055	0.297	0.308	0.065
DepthAnything	N/A	N/A	0.230	0.066	0.126
GRIN_NI	**0.048**	**0.049**	**0.198**	**0.058**	**0.061**

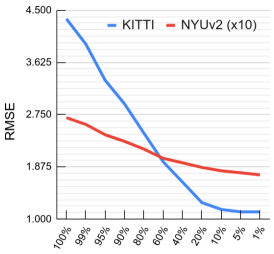
Table 2. **Zero-shot relative monocular depth estimation results (AbsRel)**. All methods use test-time scale alignment, and do not require intrinsics as input. N/A indicates methods trained on the target dataset. **GRIN_NI** indicates our model (Table 1) evaluated without intrinsics.

5.4. Zero-Shot Relative Depth Estimation

Even though our main focus is on *metric* depth estimation, here we explore how GRIN can also be applied in the context of *relative* depth estimation, where predictions are accurate up-to-scale. In this setting, camera intrinsics are not required, since the model does not need to reason over physical 3D properties of the environment, focusing instead on 2D appearance cues. Thus, we replace them with default pinhole values: $f_x = c_x = W/2$ and $f_y = c_y = H/2$, and reutilize our pre-trained metric model (Table 1). Results of this experiment are shown in Table 2, indicating that GRIN also outperforms the current state-of-the-art in relative depth estimation across multiple datasets, with the added benefit that it can also produce metric depth estimates if intrinsics are available.



(a) Qualitative examples of uncertainty maps, given by the standard deviation from multiple samples. More examples are available in the supplementary material.



(a) RMSE results with varying confidence levels.

Figure 5. **Uncertainty estimation analysis** using multiple GRIN samples. In (a), Depth and uncertainty maps are calculated taking the *median* and *standard deviation* of $s = 10$ samples. In (b) we show improvements in depth estimation by only evaluating a percentage of pixels with lower standard deviation. More examples can be found in the supplementary material.

Method	Intrinsics	*KITTI* AbsRel	*KITTI* RMSE	*NYUv2* AbsRel	*NYUv2* RMSE
Metric3D	✓	0.058	2.770	0.083	0.310
ZoeDepth	-	0.057	2.586	0.077	0.277
ZeroDepth	✓	0.053	2.087	0.074	0.269
DMD	✓	0.053	2.411	0.072	0.296
DepthAnything	-	0.046	2.180	0.056	0.264
GRIN_FT_NI	-	**0.043**	**1.953**	**0.051**	**0.251**

Table 3. **In-domain metric monocular depth estimation results**. All methods were fine-tuned on the training splits of the validation datasets. **GRIN_FT_NI** indicates our model (Table 1) fine-tuned without intrinsics.

Method	*KITTI* AbsRel	*KITTI* RMSE	*KITTI* $\delta < 1.25$	*NYUv2* AbsRel	*NYUv2* RMSE	*NYUv2* $\delta < 1.25$
A w/o local	0.057	2.624	0.941	0.079	0.301	0.944
B w/o global	0.074	2.973	0.914	0.092	0.431	0.913
C linear projection	**0.046**	**2.178**	**0.985**	0.065	0.271	0.972
D log-e projection	0.049	2.465	0.971	**0.055**	**0.198**	**0.982**
E single sample	0.048	2.498	0.973	0.061	0.258	0.971
GRIN	**0.046**	2.251	0.983	0.058	0.209	0.980

Table 4. **Ablation study** of different design choices.

5.5. Fine-Tuning Experiments

Although our main focus is on *zero-shot* depth estimation, here we explore how GRIN can also be *fine-tuned* in-domain to further improve performance in a particular setting, at the expense of generalization. Note that in this setting intrinsics are also not required (see Section 5.4) due to the absence of the *geometric domain gap*, since the model is over-fitting to a single camera geometry, and therefore can generate metric predictions without the need to reason over physical 3D properties. Results are shown in Table 3, indicating that GRIN also outperforms other metric depth estimation methods that use in-domain training data.

5.6. Ablation Study

Here we ablate different aspects and design choices of GRIN, with quantitative results in Table 4. First, we ablate the use of different forms of local and global conditioning. In (A) we show that removing image embeddings for local conditioning leads to noticeable performance degradation. We attribute this behavior to the lack of visual information for pixel-specific denoising, that now can only rely on geometric information, which is locally smooth and struggles to capture sharp discontinuities. Similarly, in (B) we show that removing global conditioning also significantly degrades performance, due to the lack of scene-level context for consistent local predictions. In (C) and (D) we explore different depth parameterizations, namely linear and natural logarithm, each emphasizing different ranges. The linear parameterization promotes more fine-grained long-range predictions, while log-e focuses on short-range predictions. Our log-10 parameterization is a compromise, producing a reasonable trade-off as evidenced by our reported numbers. In (E) we evaluate single-sample estimates, which leads to noisier predictions as shown by a higher RMSE. In Figure 5 we show uncertainty maps from multiple samples, and how these can improve depth estimation by focusing on predictions with lower uncertainty [27].

6. Conclusion

We introduce GRIN (Geometric RIN), a diffusion-based framework for depth estimation designed to circumvent two of the main shortcomings shown by recent diffusion methods when applied to this task, namely (i) the inability to properly leverage sparse training data; and (ii) the lack of specialized auto-encoders. We build upon the highly efficient and domain-agnostic RIN architecture, and modify it to include visual conditioning with 3D geometric embeddings, which enables the learning of priors anchored in physical properties. To directly ingest unstructured ground-truth supervision, we operate at a pixel-level, and introduce global conditioning as a way to preserve dense scene-level information when training with sparse labels. As a result, GRIN establishes a new state of the art in zero-shot metric monocular depth estimation, outperforming published methods that rely on large-scale image-based pre-training.

References

[1] 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. 6

[2] 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

[3] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv:2001.10773*, 2020. 6, 7

[4] 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. 2, 6, 7

[5] Ziyi Chang, George Alex Koulieris, and Hubert P. H. Shum. On the design fundamentals of diffusion models: A survey, 2023. 3

[6] Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton, and David J. Fleet. A generalist framework for panoptic segmentation of images and videos. *arXiv preprint arXiv:2210.06366*, 2022. 2

[7] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms, 2023. 6

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2, 6

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 2, 3

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

[11] Yiqun Duan, Xianda Guo, and Zheng Zhu. Diffusiondepth: Diffusion denoising approach for monocular depth estimation, 2023. 2, 3, 7

[12] 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. 2, 6

[13] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 2

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

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

[16] Ashkan Ganj, Yiqin Zhao, Hang Su, and Tian Guo. Mobile ar depth estimation: Challenges & prospects – extended version, 2023. 1

[17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 6, 7

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

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

[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] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018. 6

[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, 6, 7

[24] 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, 6

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

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

[27] Vitor Guizilini, Igor Vasiljevic, Dian Chen, Rares Ambrus, and Adrien Gaidon. Towards zero-shot scale-aware monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023. 1, 2, 4, 6, 7, 8

[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. 4

[29] Vitor Guizilini, Igor Vasiljevic, Jiading Fang, Rares Ambrus, Sergey Zakharov, Vincent Sitzmann, and Adrien Gaidon. Delira: Self-supervised depth, light, and radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023. 4

[30] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1

[31] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1, 2, 3

[32] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022. 3

[33] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours:

Self-driving motion prediction dataset. In *CoRL*, volume 155, pages 409–418, 2020. 6

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

[35] Allan Jabri, David J. Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In *ICML*, 2023. 2, 3, 4

[36] Yuanfeng Ji, Zhe Chen, Enze Xie, Lanqing Hong, Xihui Liu, Zhaoqiang Liu, Tong Lu, Zhenguo Li, and Ping Luo. Ddp: Diffusion model for dense visual prediction. *arXiv preprint arXiv:2303.17559*, 2023. 1, 2, 3, 4, 7

[37] Takayuki Kanai, Igor Vasiljevic, Vitor Guizilini, Adrien Gaidon, and Rares Ambrus. Robust self-supervised extrinsic self-calibration. In *The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023. 2

[38] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation, 2023. 1, 2, 3, 4, 7

[39] 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

[40] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 6

[41] 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

[42] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, 2023. 3, 4

[43] Xingtong Liu, Ayushi Sinha, Mathias Unberath, Masaru Ishii, Gregory D Hager, Russell H Taylor, and Austin Reiter. Self-supervised learning for dense depth estimation in monocular endoscopy. In *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, pages 128–138. Springer, 2018. 1

[44] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[45] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–421, 2020. 4

[46] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 2, 6, 7

[47] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. 2

[48] 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. 6

[49] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 6, 7

[50] 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. 2

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

[52] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*. Springer, 2015. 1, 3

[53] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *SIGGRAPH*, 2022. 4

[54] Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J. Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1, 2, 3, 4, 7

[55] Saurabh Saxena, Junhwa Hur, Charles Herrmann, Deqing Sun, and David J. Fleet. Zero-shot metric depth with a field-of-view conditioned diffusion model, 2023. 1, 2, 3, 4, 6, 7

[56] Saurabh Saxena, Abhishek Kar, Mohammad Norouzi, and David J. Fleet. Monocular depth estimation using diffusion models, 2023. 2, 3, 7

[57] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *ECCV*, 2020. 2

[58] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 3

[59] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[60] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, October 2020. 6

[61] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015. 6, 7

[62] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2

[63] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya

Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014. 5

[64] 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. 2, 6

[65] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3

[66] Igor Vasiljevic, Nick 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. 6, 7

[67] 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. 4

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

[69] 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. 2, 6

[70] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth. In *CVPR*, 2021. 2

[71] 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. 2

[72] 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 *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021. 6

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

[74] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data, 2024. 2, 3, 4, 7

[75] 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. 4

[76] 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 *ICCV*, 2023. 2, 6, 7

[77] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Simon Chen, Yifan Liu, and Chunhua Shen. Towards accurate reconstruction of 3d scene shape from a single monocular image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[78] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. New crfs: Neural window fully-connected crfs for monocular depth estimation, 2022. 6

[79] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 4

[80] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. *ICCV*, 2023. 2

[81] Lipu Zhou, Jiamin Ye, Montiel Abello, Shengze Wang, and Michael Kaess. Unsupervised learning of monocular depth estimation with bundle adjustment, super-resolution and clip loss. *arXiv:1812.03368*, 2018. 2

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