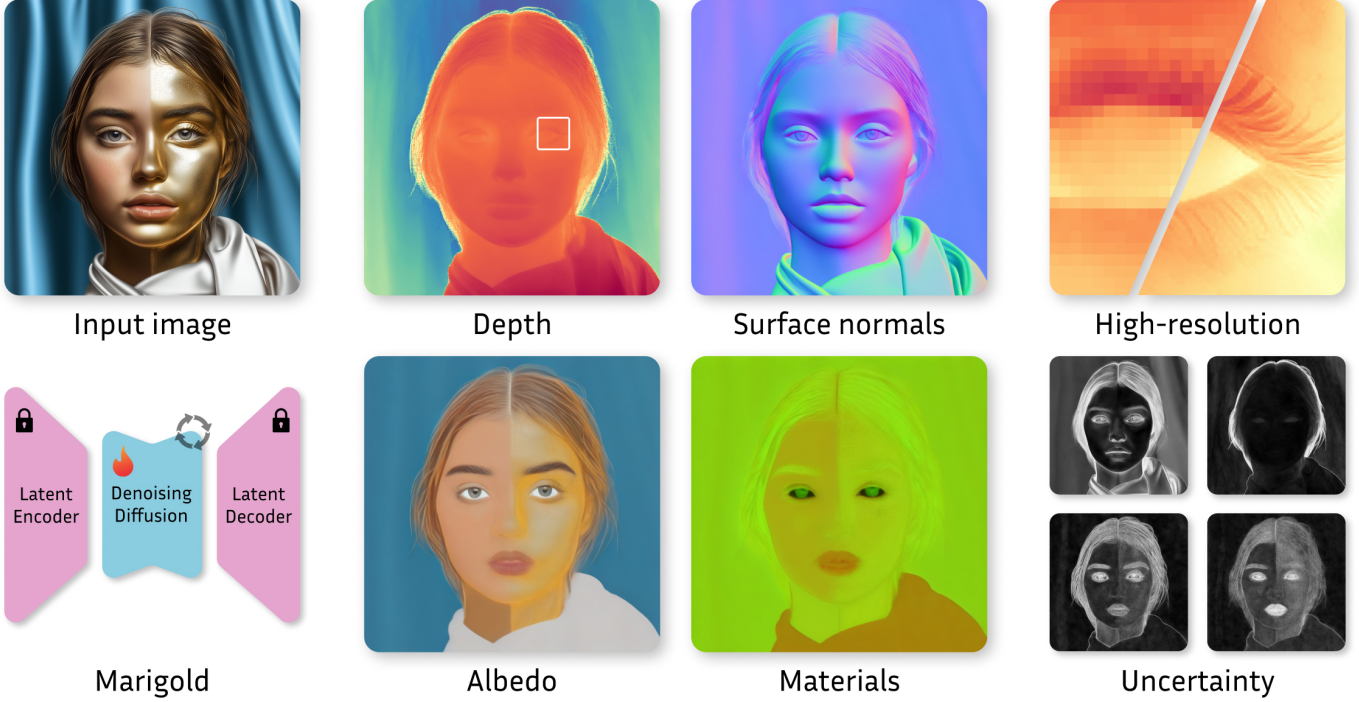


# Marigold: Affordable Adaptation of Diffusion-Based Image Generators for Image Analysis

Bingxin Ke\*, Kevin Qu\*, Tianfu Wang\*, Nando Metzger\*, Shengyu Huang, Bo Li,  
Anton Obukhov<sup>†,\*</sup>, Konrad Schindler<sup>†</sup>



We present Marigold, a fine-tuning protocol for various image analysis tasks, and a family of associated diffusion models. Without loss of generality, these include monocular depth estimation, surface normals prediction, and intrinsic image decomposition. Its core principle is to leverage the rich visual knowledge stored in modern generative image models. As a generative model derived from Stable Diffusion and fine-tuned with synthetic data, Marigold can zero-shot transfer to unseen datasets, offering state-of-the-art results. The visualizations above demonstrate the strong out-of-distribution performance: without observing a single image other than synthetic rooms and dashboard views, Marigold can extract pixel-perfect depth maps, surface normals, and intrinsic decomposition of images, ready for downstream tasks.

**Abstract**—The success of deep learning in computer vision over the past decade has hinged on large labeled datasets and strong pretrained models. In data-scarce settings, the quality of these pretrained models becomes crucial for effective transfer learning. Image classification and self-supervised learning have traditionally been the primary methods for pretraining CNNs and transformer-based architectures. Recently, the rise of text-to-image generative models, particularly those using denoising diffusion in a latent space, has introduced a new class of foundational models trained on massive, captioned image datasets. These models’ ability to generate realistic images of unseen content suggests they possess a deep understanding of the visual world. In this work, we present **Marigold**, a family of conditional generative models and a fine-tuning protocol that extracts the knowledge from pretrained latent diffusion models like Stable Diffusion and adapts them for dense image analysis tasks, including monocular depth estimation, surface normals prediction, and intrinsic decomposition. Marigold requires minimal modification

of the pre-trained latent diffusion model’s architecture, trains with small synthetic datasets on a single GPU over a few days, and demonstrates state-of-the-art zero-shot generalization. Project page: <https://marigoldcomputervision.github.io>.

**Index Terms**—Denoising diffusion, image analysis, image generation, foundational models, transfer learning.

## I. INTRODUCTION

THE introduction of ImageNet [1] laid the foundation for training deep Convolutional Neural Networks (CNNs), such as AlexNet [2], catalyzing further advances in the computer vision field: in data acquisition, neural architectures, and training techniques. With the advent of VGG [3] and ResNet [4] architectures, transfer learning [5] became essential for training high-performance computer vision models and reducing training time of semantic segmentation [6], depth prediction [7], and other downstream tasks. In many cases, training a neural network from random weight initialization is

Work done at the Photogrammetry and Remote Sensing Laboratory, ETH Zürich, Switzerland. \* denotes equal technical contribution. † denotes equal supervision. Corresponding author: Konrad Schindler (schindler@ethz.ch).

claimed not feasible [6]. Modern deep learning frameworks [8] have since made it easy to use pretrained models by allowing practitioners to load pretrained weights with a simple setting like `pretrained=True` during model creation.

The rise of large text-to-image generative models [9] and denoising diffusion approaches [10], [11] has opened new opportunities for leveraging the rich priors embedded in foundational models. A breakthrough in this area came with the introduction of Latent Diffusion Models (LDMs), a class of models exemplified by the widely known Stable Diffusion (SD) [12]. These models operate in the compressed latent space of a pretrained Variational Autoencoder (VAE), enabling significant resource savings in both training and inference. Trained on the internet-scale LAION-5B dataset of captioned images [13], Stable Diffusion excels in realism and diversity. Its open-source availability, low computational requirements for inference, and integration with toolkits like `diffusers` [14] have enabled widespread experimentation by researchers and artists. The abundance of customization recipes [15]–[17] has prompted many notable extensions that focus on enhancing the controllability of the original image generation task.

Repurposing text-to-image LDMs from image generation to image analysis is a recent development in generative imaging. The motivation is simple: if a diffusion model demonstrates a deep understanding of the visual world through high-quality image generation, that same understanding can be leveraged to derive a versatile regression model for image analysis. To this end, in our recent work [18], we introduced **Marigold-Depth**, an LDM-based state-of-the-art zero-shot affine-invariant monocular depth estimator, along with a simple and resource-efficient fine-tuning protocol for Stable Diffusion.

Marigold-Depth proposed several key novelties unlocking the potential of LDMs for image understanding:

- (1) reusing the LDM’s VAE to encode not just the input image but also the output modality into the latent space;
- (2) using only high-quality synthetic data;
- (3) short resource-efficient fine-tuning protocol;
- (4) generative modeling of a conditional distribution rather than predicting its mode as end-to-end approaches do.

Some of these properties are organically entangled. (1 $\leftrightarrow$ 2): Encoding the modality into latent space is only possible when it is noise-free and pixel-complete – rarely the case with the real depth ground truth. (2 $\leftrightarrow$ 3): A short fine-tuning protocol preserves prior knowledge. It requires diverse, consistently labeled, and noise-free data to reduce noise in weight updates, which are satisfiable with synthetic data. (1 $\leftrightarrow$ 3): Operation in latent space ensures affordable fine-tuning and inference on a single consumer Graphics Processing Unit (GPU), empowering research even outside large labs.

The importance of synthetic data and strong prior for depth estimation have been subsequently confirmed in Depth Anything V2 [19]. Although their end-to-end model achieves impressive performance in zero-shot benchmarks, it involves a 3-stage training procedure, a teacher-student separation, and generating 62M pseudo labels; both do not fit the bill of a simple and affordable transfer learning recipe.

For the property (4), as shown in [18], modeling the distribution of depth conditioned on the input image with

an LDM allows for multiple plausible interpretations of the input. This ability is essential for solving ill-posed problems, as there may be no single correct output due to over-exposed input, blur, ambiguities of transparent objects, *etc.* Obtaining samples from the conditional distribution with Marigold is as simple as starting the diffusion process from different noise samples given the same input. Multiple such predictions can be ensembled to approximate the mode of a conditional distribution. Ensembling is required to evaluate prediction quality in standard benchmarks comprised of image-depth ground truth pairings, an established evaluation protocol [20], [21]. Additionally, computing predictive uncertainty becomes tractable given such an ensemble.

The multi-step inference and the computational redundancy of the ensembling typically result in many function evaluations (NFEs), which slow down inference speed – initially, a major point of criticism of the original Marigold-Depth [18]. To this end, we explored Latent Consistency Distillation [22] to reduce the number of sampling steps arbitrarily low, even to just one. Simultaneously, several works explored other techniques to bring sampling steps down. Some preserved the generative nature of the model [23]–[25]. Additionally, [23] and [26] showed the possibility of scoring high in the said benchmarks by re-casting Marigold as an end-to-end network, effectively yielding just one function evaluation. In this paper, we continue focusing on the generative formulation.

Notably, Garcia *et al.* [23] discovered sub-optimal settings in the diffusion scheduler of the original Marigold-Depth [18] and proposed a correction, leading to significant improvement of that exact model’s performance in the same benchmarks in the few-step inference regime. We gratefully incorporated this observation and integrated this regime into our study. With single-step inference and technical enhancements such as using a lightweight compatible VAE [27] and low-precision weight quantization, Marigold can now produce predictions in under 100ms on most commodity hardware.

In this paper, we follow up on the initial body of work [18], recap it in Sec. III, and extend our study with several dense image analysis tasks [28] having a long history in computer graphics: surface normals prediction in Sec. IV and intrinsic image decomposition in Sec. V. Additionally, we introduce a protocol for distilling Marigold models into Latent Consistency Models (LCMs) in Sec. VI and the new High-Resolution (HR) model and inference strategy in Sec. VII.

To summarize, our contributions are:

- (1) *Marigold* – a simple and resource-efficient fine-tuning protocol to convert a foundational LDM image generator into a zero-shot generative image analysis model with robust in-the-wild generalization capability. Training a Marigold takes less than 3 GPU-days and can be accomplished with most commodity hardware;
- (2) A comprehensive study of training diffusion models with Marigold for monocular depth estimation, surface normal prediction, and intrinsic image decomposition tasks;
- (3) Fast (sub-100ms) few- or single-step inference;
- (4) Overcoming the resolution bias of the base model, enabling high-resolution inference.



## II. RELATED WORK

### A. Diffusion Models

Denoising Diffusion Probabilistic Models (DDPMs) [10], [29] generate data by reversing a Gaussian noise diffusion process. Denoising Diffusion Implicit Models (DDIMs) [30] extend this by introducing a non-Markovian shortcut for faster sampling at inference time. Latent Consistency Models (LCMs) [22], [31] distill DDPM models into consistency functions that map points on the diffusion ODE trajectory [32] to the same output, thereby introducing a different parameterization and reducing inference time. In the realm of image generation, Rombach *et al.* [12] have revolutionized generative modeling with their Stable Diffusion model, trained with LAION-5B [13] dataset of 2.3B text-image pairs. The cornerstone of their approach is an LDM, where the denoising process is run in an efficient latent space, drastically reducing the complexity of the learned mapping. This model holds rich visual priors, providing a solid foundation for our generalizable image analysis models and a base for our transfer learning approach.

### B. Foundation Models

Vision Foundation Models (VFM) are large neural networks trained on internet-scale data. The extreme scaling leads to the emergence of high-level visual understanding, such that the model can then be used as is [33] or fine-tuned to a wide range of downstream tasks with minimal effort [34]. Prompt tuning methods [35]–[37] can efficiently adapt VFMs towards dedicated scenarios by designing suitable prompts. Feature adaptation methods [38]–[42] can further pivot VFMs towards different tasks. Direct tuning enables more flexible adaptation, especially in few-shot customization scenarios like DreamBooth [17]. As we show in this paper, Marigold can be interpreted as an instance of this type of tuning, where Stable Diffusion [12] plays the role of the foundation model for multiple image analysis tasks. Our models exhibit strong in-the-wild performance thanks to the foundational prior and efficient fine-tuning protocol (*cf.* the teaser figure).

### C. Monocular Depth Estimation

The pioneering work [7] introduced an end-to-end trainable network and showed that metric depth for a dataset can be recovered with a single sensor. Successive improvements have come from various fronts, including various parameterizations (ordinals, bins, planar maps, CRFs, *etc.*) [43]–[50] and switching CNN backbones to vision transformers [51]–[54]. A class of works [55]–[57] relies on privileged information fed alongside input, such as camera intrinsics. Estimating depth “in the wild” or “out-of-distribution” refers to methods with robustness across a wide range of possibly unfamiliar settings where no privileged information is available. MegaDepth [58] and DiverseDepth [59] utilize extensive internet photo collections to train models that can adapt to unseen data, while MiDaS [60] achieves generality by training on a mixture of multiple datasets. To unify representations across datasets, MiDaS estimates affine-invariant depth – up to unknown shift and scale. The step from CNNs to vision transformers has further boosted

performance, as evidenced by DPT (MiDaS v3) [61]. Depth Anything [62] took data scaling to the next level by relying on DINOv2 [63] foundational model prior trained on 142M images with self-supervision and subsequent training with 62M pseudo-labels, 1M real depth annotations, and 0.5M synthetic ones. Several methods [42], [42], [64], [65] proposed using DDPMs and LDMs for depth estimation. However, they either train models from scratch, use Stable Diffusion [12] as a feature extractor, resort to custom latent spaces, operate in pixel space, or require extensive training. Our previous work [18] proposed fine-tuning a generative text-to-image LDM such as Stable Diffusion, trained with LAION-5B [13], a dataset of 2.3B image-text pairs, towards affine-invariant depth using just 74K samples from the HyperSim [66] and Virtual KITTI [67] synthetic datasets. Marigold demonstrated impressive zero-shot generalization both in benchmarks and “in the wild”.

Since uploading [18], Depth Anything V2 [19] confirmed our findings about the role of synthetic data in the task and retired real data from their pipeline, achieving impressive performance gains. E2E-FT [23] and GenPercept [26] performed end-to-end fine-tuning. The former also proposed a fix for a few-step DDIM scheduler inference regime. E2E-FT also demonstrated that end-to-end networks can score higher in zero-shot benchmarks than the similar generative model. Two more works kept the generative nature of the base model: Lotus [25] switched to predicting the target using exactly one step; DepthFM [24] adopted flow matching [68] at training. GeoWizard [69] proposed to estimate the depth and surface normals jointly, although using privileged information about scene type. BetterDepth [70] introduced a Marigold-based refiner for coarse depth inputs. SteeredMarigold [71] used Marigold as a generative prior in order to perform depth densification. Robustness under challenging conditions [72] was tackled through depth-conditioned generation of training data, an approach similar to DGINStyle [73]. ChronoDepth [74] and DepthCrafter [75] address the temporal consistency of depth prediction in the video domain by performing Marigold-like fine-tuning of video diffusion models.

We stick to the generative formulation of Marigold, incorporate the DDIM fix [23], and recap the method [18] (Sec. III).

### D. Monocular Surface Normals Prediction

Early monocular surface normals estimation methods often employed CNNs consisting of a feature extractor backbone followed by a prediction head [76]–[79]. Over time, various improvements have been proposed. Bae *et al.* [80] suggested estimating aleatoric uncertainty and using uncertainty-guided sampling during training to enhance prediction quality for small structures and object boundaries. Omnidata v2 [81] introduced a transformer-based model trained on 12M images, applying sophisticated 3D data augmentation and enforcing cross-task consistency. DSINE [82] identified and incorporated inductive biases tailored for surface normals estimation. It leverages the per-pixel ray direction coupled with a ray-based activation function and learns the relative rotation between neighboring surface normals.

Since uploading [18], several papers have repurposed diffusion models to surface normals estimation. GeoWizard [69]

jointly estimates depth and surface normals, although it uses privileged information about scene type. Shortly after, we released Marigold-Normals v0.1, a preview model for monocular surface normals estimation trained similarly to Marigold-Depth, albeit just on HyperSim. GenPercept [26] treats the denoising U-Net as a deterministic backbone and employs one-step inference. StableNormal [83] aims to reduce the inherent stochasticity of diffusion models by using a two-stage coarse-to-fine strategy. A single-step surface normals estimator first produces an initial coarse estimate, followed by a refinement process that recovers finer geometric details, semantically guided by DINOv2 [63] features. Lotus [25] directly predicts annotations instead of noise and reformulates the multi-step diffusion into a single-step procedure. Garcia *et al.* [23] proposed a single-step inference fix for Marigold’s DDIM scheduler to enhance inference speed. Additionally, they reframe a single-step generative predictor into an end-to-end network based on the same architecture.

In what follows, we continue developing the ideas behind Marigold-Normals and arrive at v1.1. We demonstrate that a careful curation of synthetic fine-tuning datasets together with the vanilla Marigold protocol without any other foundational models, multi-task aggregation, privileged information, or refinement stages achieves the best performance in most evaluation datasets, compared to the other recent diffusion-based surface normals estimation methods (Sec. IV).

### E. Intrinsic Image Decomposition (IID)

Intrinsic image decomposition aims to recover the intrinsic properties of objects in an image, including albedo (surface reflectance), shading, and Bidirectional Reflectance Distribution Function (BRDF) parameters, such as roughness and metallicity. It was introduced by Horn [28] and later studied by Barrow *et al.* [84]. The theory evolved from early prior-based approaches, such as the retinex theory, to modern deep learning-based methods. First, deep learning approaches typically utilized a feed-forward convolutional network [85]–[88] or a transformer [89] to predict pixel-level intrinsic decomposition channels from the input image. Careaga *et al.* [90] proposed a multi-step approach that first predicts initial albedo and grayscale shading maps using an off-the-shelf network [91], and then progressively refines them. With the recent advent of vision foundation models, especially generative models, alternative solutions have shown success by utilizing StyleGAN [92], [93] or diffusion models [94]–[97]. DMP [95] learns a deterministic mapping between an input and the IID task (albedo and shading) through a low-rank adaptation of text-to-image models. IID-Diffusion [96] uses a custom latent encoder and CLIP [98] features to guide the fine-tuned Stable Diffusion [12] on InteriorVerse [99] decomposition into albedo and material properties: roughness and metallicity. RGB $\leftrightarrow$ X [97] learns a bijection between input images and various modalities, including the ones used in IID-Diffusion. Their model, based on the pretrained Stable Diffusion, has a similar architecture to Marigold, yet uses the text encoder to switch between pre-defined modalities.

We train two Marigold-IID models: one that predicts albedo, roughness, and metallicity, and another that estimates albedo,

non-diffuse shading, and a residual diffuse component. We compare our models to IID-Diffusion [96], RGB $\leftrightarrow$ X [97] and Careaga *et al.* [90] in their respective domains on the InteriorVerse [99] and HyperSim [66] datasets. Our simpler models achieve highly competitive performance in both quantitative and qualitative evaluations (Sec. V).

### F. High-Resolution Estimation

Models fine-tuned from Stable Diffusion [12] usually exhibit resolution bias towards the original resolution used to train the text-to-image LDM. The same applies to all Marigold models: their processing resolution defaults to the recommended value of 768 inherited from the base model, which should correspond to the longest side of the input image. As a result, large resolutions suffer a major loss of details during downsampling to the processing resolution and upsampling output to the original size (see the teaser figure). This issue prompted us to investigate approaches to high-resolution inference with Marigold. Although the study is centered around Marigold-Depth, the findings apply to other modalities.

BoostingDepth [100] fuses local patches into a global canvas through an additional GAN network. PatchFusion [101] introduces a tile-based framework that combines globally consistent coarse features with finer local features using a patch-wise fusion network. Moreover, it ensures consistency across patches during training without post-processing. PatchRefiner [102] performs high-resolution depth estimation by refining predictions with a pseudo-labeling strategy. MultiDiffusion [103] is an inference protocol for pre-trained diffusion models that generates high-resolution outputs by performing diffusion over overlapping tiles in an interleaved fashion. While it has been used in generative tasks like semantic segmentation [73], it has not been applied yet to enhance the resolution of image analysis tasks. Depth Pro [104] proposed concurrently a multi-scale vision transformer trained on tens of datasets, capable of predicting metric depth for HD images in a split second.

Our Marigold-HR inference strategy attains competitive or better performance compared to these other methods (Sec. VII).

## III. BASE MODEL: MARIGOLD-DEPTH

In this section, we recap the details of our fine-tuning protocol in the context of monocular depth estimation [18].

### A. Generative Formulation

We pose monocular depth estimation as a conditional denoising diffusion generation task and train Marigold to model the conditional distribution  $D(\mathbf{d} \mid \mathbf{x})$  over depth  $\mathbf{d} \in \mathbb{R}^{W \times H}$ , where the condition  $\mathbf{x} \in \mathbb{R}^{W \times H \times 3}$  is an RGB image.

In the *forward* process, which starts at  $\mathbf{d}_0 := \mathbf{d}$  from the conditional distribution, Gaussian noise is gradually added at levels  $t \in \{1, \dots, T\}$  to obtain noisy samples  $\mathbf{d}_t$  as

$$\mathbf{d}_t = \sqrt{\bar{\alpha}_t} \mathbf{d}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon} \quad (1)$$

where  $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ ,  $\bar{\alpha}_t := \prod_{s=1}^t 1 - \beta_s$ , and  $\{\beta_1, \dots, \beta_T\}$  is the variance schedule of a process with  $T$  steps. In the *reverse* process, the denoising model  $\epsilon_\theta(\cdot)$  parameterized with learned parameters  $\theta$  gradually removes noise from  $\mathbf{d}_t$  to obtain  $\mathbf{d}_{t-1}$ .

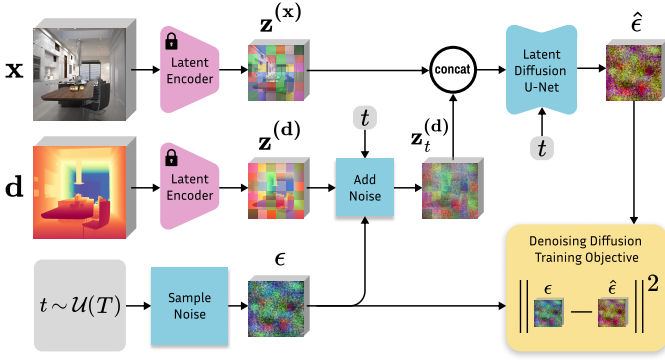


Fig. 1: **Overview of the Marigold fine-tuning protocol.** Starting from a pretrained Stable Diffusion, we encode the image  $\mathbf{x}$  and depth  $\mathbf{d}$  into the latent space using the original Stable Diffusion VAE. We fine-tune just the U-Net by optimizing the standard diffusion objective relative to the depth latent code. Image conditioning is achieved by concatenating the two latent codes before feeding them into the U-Net. The first layer of the U-Net is modified to accept concatenated latent codes. See details in Sec. III-B and Sec. III-C.

At training time, parameters  $\theta$  are updated by taking a data pair  $(\mathbf{x}, \mathbf{d})$  from the training set, noising  $\mathbf{d}$  with sampled noise  $\epsilon$  at a random timestep  $t$ , computing the noise estimate  $\hat{\epsilon} = \epsilon_\theta(\mathbf{d}_t, \mathbf{x}, t)$  and minimizing one of the denoising diffusion objective functions. The canonical standard noise objective  $\mathcal{L}$  is given as follows [29]:

$$\mathcal{L} = \mathbb{E}_{\mathbf{d}_0, \epsilon \sim \mathcal{N}(0, I), t \sim \mathcal{U}(T)} \|\epsilon - \hat{\epsilon}\|_2^2. \quad (2)$$

At inference time,  $\mathbf{d} := \mathbf{d}_0$  is reconstructed starting from a normally-distributed variable  $\mathbf{d}_T$ , by iteratively applying the learned denoiser  $\epsilon_\theta(\mathbf{d}_t, \mathbf{x}, t)$ .

We consider the latent space formed in the bottleneck of a VAE, trained independently of the denoiser, to enable latent space compression and perceptual alignment with the data space. To translate our formulation into the latent space, for a given depth map  $\mathbf{d}$ , the corresponding latent code is given by the encoder  $\mathcal{E}: \mathbf{z}^{(d)} = \mathcal{E}(\mathbf{d})$ . Given a depth latent code, a depth map can be recovered with the decoder  $\mathcal{D}: \hat{\mathbf{d}} = \mathcal{D}(\mathbf{z}^{(d)})$ . The conditioning image  $\mathbf{x}$  is also naturally translated into the latent space as  $\mathbf{z}^{(x)} = \mathcal{E}(\mathbf{x})$ . The denoiser is henceforth trained in the latent space:  $\epsilon_\theta(\mathbf{z}_t^{(d)}, \mathbf{z}^{(x)}, t)$ . The adapted inference procedure involves one extra step – the decoder  $\mathcal{D}$  reconstructing the data  $\hat{\mathbf{d}}$  from the estimated clean latent  $\mathbf{z}_0^{(d)}: \hat{\mathbf{d}} = \mathcal{D}(\mathbf{z}_0^{(d)})$ .

### B. Network Architecture

We base our model on a pretrained text-to-image LDM Stable Diffusion v2 [12]. With minimal changes to the model, we turn it into a conditional depth map generator (Fig. 1).

**Depth encoder and decoder.** We take the frozen VAE to encode *both* the image and its corresponding depth map into a latent space for training our conditional denoiser. Given that the encoder, which is designed for 3-channel (RGB) inputs, receives a single-channel depth map, we replicate the depth map into three channels to simulate an RGB image. At this point, the

data range of the depth data plays a significant role in enabling affine-invariance. We discuss our normalization approach in Sec. III-C. We verified that without any modification of the VAE or the latent space structure, the depth map can be reconstructed from the encoded latent code with a negligible error, *i.e.*,  $\mathbf{d} \approx \mathcal{D}(\mathcal{E}(\mathbf{d}))$ . This is the first check to be performed when following the Marigold fine-tuning protocol for a new modality. At inference time, the depth latent code is decoded once at the end of diffusion, and the average of three channels is taken as the predicted depth map. Extending Marigold to another modality with a different number of channels may prompt allocating new latent space for each triplet of channels.

**Adapted denoising U-Net.** To implement the conditioning of the latent denoiser  $\epsilon_\theta(\mathbf{z}_t^{(d)}, \mathbf{z}^{(x)}, t)$  on input image  $\mathbf{x}$ , we concatenate the image and depth latent codes into a single input  $\mathbf{z}_t = \text{cat}(\mathbf{z}_t^{(d)}, \mathbf{z}^{(x)})$  along the feature dimension. The input channels of the latent denoiser are then doubled to accommodate the expanded input  $\mathbf{z}_t$ . To prevent inflation of activations magnitude of the first layer and keep the pre-trained structure as faithfully as possible, we duplicate the weight tensor of the input layer and divide its values by two. A similar conditioning mechanism, except for the zero weights initialization, was previously used in InstructPix2Pix [15].

### C. Fine-Tuning for Depth Estimation

**Affine-invariant depth normalization.** For the ground truth depth maps  $\mathbf{d}$ , we implement a linear normalization such that the depth primarily falls in the value range  $[-1, 1]$ , to match the designed input value range of the VAE. Such normalization serves two purposes. First, it is the convention for working with the original Stable Diffusion VAE. Second, it enforces a canonical affine-invariant depth representation independent of the data statistics – any scene must be bounded by near and far planes with extreme depth values. The normalization is achieved through an affine transformation computed as

$$\tilde{\mathbf{d}} = \left( \frac{\mathbf{d} - \mathbf{d}_2}{\mathbf{d}_{98} - \mathbf{d}_2} - 0.5 \right) \times 2, \quad (3)$$

where  $\mathbf{d}_2$  and  $\mathbf{d}_{98}$  correspond to the 2% and 98% percentiles of individual depth maps. This normalization allows Marigold to focus on pure affine-invariant depth estimation.

**Training on synthetic data.** Real depth datasets suffer from missing depth values caused by the physical constraints of the capture rig and the physical properties of the sensors. Specifically, the disparity between cameras and reflective surfaces diverting LiDAR laser beams are inevitable sources of ground truth noise and missing pixels [105], [106]. In contrast to prior work that utilized diverse real datasets to achieve generalization [60], [107], we train exclusively with synthetic depth datasets. As with the depth normalization rationale, this decision has two objective reasons. First, synthetic depth is inherently dense and complete, meaning that every pixel has a valid ground truth depth value, allowing us to feed such data into the VAE, which can not handle data with invalid pixels. Second, synthetic depth is the cleanest possible form of depth, which is guaranteed by the rendering pipeline. It provides the cleanest examples and reduces noise in gradient updates during



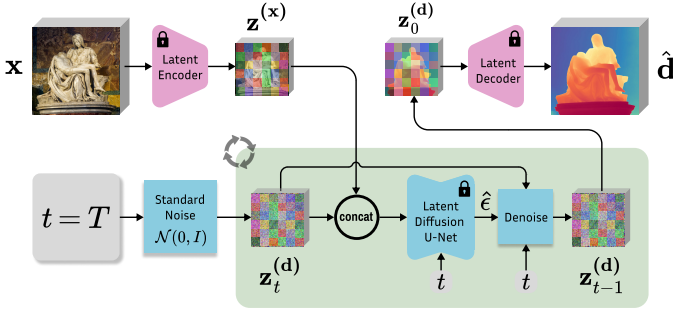


Fig. 2: **Overview of the Marigold inference scheme.** Given an image  $x$ , we encode it with the original Stable Diffusion VAE into the latent code  $z^{(x)}$ , and concatenate with the depth latent  $z_t^{(d)}$  before giving it to the modified fine-tuned U-Net on every denoising iteration. After executing the schedule of  $T$  steps, the resulting depth latent  $z_0^{(d)}$  is decoded into an image whose 3 channels are averaged to get the final estimation  $\hat{d}$ . See Sec. III-D for details.

the short fine-tuning protocol. Thus, the remaining concern is the sufficient diversity or domain gaps between synthetic and real data, which sometimes limits generalization ability. As demonstrated in our experiments across modalities, our choice of synthetic datasets leads to impressive zero-shot transfer.

#### D. Inference

**Latent diffusion denoising.** The overall inference pipeline is presented in Fig. 2. We encode the input image into the latent space, initialize depth latent as standard Gaussian noise, and progressively denoise it with the same schedule as during fine-tuning. We use DDIM [30] to perform non-Markovian sampling with re-spaced steps for accelerated inference. The final depth map is decoded from the latent code using the VAE decoder and postprocessed by averaging channels.

**Test-time ensembling.** The stochastic nature of the inference pipeline leads to varying predictions depending on the initialization noise in  $z_T^{(d)}$ . Capitalizing on that, we propose the following test-time ensembling scheme, capable of combining multiple inference passes over the same input. For each input sample, we can run inference  $N$  times. To aggregate these affine-invariant depth predictions  $\{\hat{d}_1, \dots, \hat{d}_N\}$ , we jointly estimate the corresponding scale  $\hat{s}_i$  and shift  $\hat{t}_i$ , relative to some canonical scale and range, in an iterative manner. The proposed objective minimizes the distances between each pair of scaled and shifted predictions  $(\hat{d}'_i, \hat{d}'_j)$ , where  $\hat{d}' = \hat{d} \times \hat{s} + \hat{t}$ . In each optimization step, we calculate the merged depth map  $\mathbf{m}$  by the taking pixel-wise median  $\mathbf{m}(x, y) = \text{median}(\hat{d}'_1(x, y), \dots, \hat{d}'_N(x, y))$ . An extra regularization term  $\mathcal{R} = |\min(\mathbf{m})| + |1 - \max(\mathbf{m})|$ , is added to prevent collapse to the trivial solution and enforce the unit scale of  $\mathbf{m}$ . Thus, the objective function can be written as follows:

$$\min_{\substack{s_1, \dots, s_N \\ t_1, \dots, t_N}} \left( \sqrt{\frac{1}{b} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \|\hat{d}'_i - \hat{d}'_j\|_2^2} + \lambda \mathcal{R} \right) \quad (4)$$

where the binominal coefficient  $b = \binom{N}{2}$  represents the number of possible combinations of image pairs from  $N$  images. After the iterative optimization for spatial alignment, the merged depth  $\mathbf{m}$  is taken as our ensembled prediction. Note that this ensembling step requires no ground truth for aligning independent predictions. This scheme enables a flexible trade-off between computation efficiency and prediction quality by choosing  $N$  accordingly.

#### E. Implementation

We use Stable Diffusion v2 [12] as base LDM, following the original pre-training setup with a  $v$ -objective [108]. We disable text conditioning and perform steps outlined in Sec. III-B. During training, we apply the DDPM noise scheduler [29] with 1000 diffusion steps. At inference time, we apply DDIM scheduler [30] and sample between 1 and 50 steps. To approximate the mode of conditional distribution and increase quality, we ensemble 10 predictions from different initial noise. Training takes 18K iterations using a batch size of 32. To fit one GPU, we use a real batch size of 2 and accumulate gradients 16 times. We use the Adam optimizer with a  $3 \cdot 10^{-5}$  learning rate. Additionally, we apply random horizontal flipping augmentation to the training data. Training our method to convergence takes approximately 2.5 days on a single Nvidia RTX 4090 GPU card. Unlike the  $10 \times 50$  zero-shot evaluation protocol, inference with one ensemble member and one diffusion step  $1 \times 1$  produces sufficiently good results fast, often sharper than the ensembled prediction. Coupled with model weight quantization and smaller compatible VAEs, such as TAESD [27], the  $1 \times 1$  prediction takes less than 100ms on most hardware.

#### F. Evaluation

**Training datasets.** We employ two synthetic datasets covering both indoor and outdoor scenes. **HyperSim** [66] is a photorealistic dataset with 461 indoor scenes. We use the official split for training, with around 54K samples from 365 scenes, filtering out incomplete samples. RGB images and depth maps are resized to  $480 \times 640$  resolution. Depth is normalized with the dataset statistics. Additionally, we transform distances relative to the focal point into depth values relative to the focal plane. The second dataset, **Virtual KITTI** [67], is a synthetic street-scene dataset featuring 5 scenes with diverse weather and camera perspectives. We crop the images to the KITTI resolution [111] and set the far plane to 80 meters. New in Marigold-Depth v1.1: (1) the training data is augmented with flipping, blurring, and color jitter; (2) DDIM timesteps are set to “trailing” and zero SNR is enabled [23] before fine-tuning.

**Evaluation datasets.** We evaluate Marigold-Depth on 5 real datasets not seen during training. **NYUv2** [21] and **ScanNet** [112] are both indoor scene datasets captured with an RGB-D Kinect sensor. For NYUv2, we utilize the designated test split, comprising 654 images. In the case of the ScanNet dataset, we randomly sampled 800 images from the 312 official validation scenes for testing. **KITTI** [111] is a street-scene dataset with sparse metric depth captured by a LiDAR sensor. We employ the Eigen test split [7] made of 652 images.

TABLE I: **Quantitative comparison** of Marigold-Depth with SOTA affine-invariant depth estimators on several zero-shot benchmarks<sup>1</sup>. In the 1<sup>st</sup> section, we list methods citing our approach [18] as well as methods that require more than 2M samples for fine-tuning (denoted by gray). We compare methods from the 2<sup>nd</sup> section with flavors of Marigold-Depth v1.0 (ours, CVPR’2024) from the 3<sup>rd</sup> section and Marigold-Depth v1.1 (ours, this paper) in the 4<sup>th</sup> section. Legend: All metrics are presented in percentage terms; bold numbers are the best, underscored second best; NFEs is the number of function evaluations required to obtain the prediction – *ensemble*  $\times$  *steps* for diffusion models and 1 for end-to-end networks. Marigold outperforms other methods in this low-data regime on indoor and outdoor scenes without access to real depth samples.

Method	NFEs	Prior	Data				NYUv2		KITTI		ETH3D		ScanNet		DIODE	
			Prior	Generated	Real	Synthetic	AbsRel ↓	$\delta 1 \uparrow$	AbsRel ↓	$\delta 1 \uparrow$	AbsRel ↓	$\delta 1 \uparrow$	AbsRel ↓	$\delta 1 \uparrow$	AbsRel ↓	$\delta 1 \uparrow$
Omnidata [107]	1	ImageNet	14M	—	12M	310K	7.4	94.5	14.9	83.5	16.6	77.8	7.5	93.6	33.9	74.2
Metric3D [55] <sup>2</sup>	1	ImageNet	14M	—	8M	—	5.8	96.3	5.3	96.5	6.4	96.5	7.4	94.2	21.1	82.5
Metric3D v2 [56] <sup>2</sup>	1	DINOv2	142M	—	16M	91K	4.3	98.1	4.4	98.2	4.2	98.3	<u>2.2</u>	<u>99.4</u>	13.6	89.5
DepthAnything [62]	1	DINOv2	142M	62M	1M	524K	4.3	98.1	7.6	94.7	12.7	88.2	4.2	98.0	27.7	75.9
DepthAnything v2 [19]	1	DINOv2	142M	62M	—	595K	4.4	97.9	7.5	94.8	13.2	86.2	—	—	6.5	95.4
DepthFM [24]	? $\times$ 1	SD v2.1	2.3B	—	—	63K	6.5	95.6	8.3	93.4	—	—	—	—	22.5	80.0
GeoWizard [69] <sup>2</sup>	10 $\times$ 50	SD v2.0	2.3B	—	51K	227K	5.2	96.6	9.7	92.1	6.4	96.1	6.1	95.3	29.7	79.2
GenPercept [26]	1	SD v2.1	2.3B	—	—	74K	5.6	96.0	9.9	90.4	6.2	95.8	—	—	35.7	75.6
Lotus-G [25]	1 $\times$ 1	SD v2.0	2.3B	—	—	59K	5.4	96.6	11.3	87.7	6.2	96.1	6.0	96.0	—	—
Lotus-D [25]	1	SD v2.0	2.3B	—	—	59K	5.3	96.7	9.3	92.8	6.8	95.3	6.0	96.3	—	—
E2E-FT [23]	1	Marigold	2.3B	—	—	74K	5.2	96.6	9.6	91.9	6.2	95.9	5.8	96.2	30.2	77.9
E2E-FT [23]	1	SD v2.0	2.3B	—	—	74K	5.4	96.5	9.6	92.1	6.4	95.9	5.8	96.5	30.3	77.6
DiverseDepth [59]	1	ImageNet	1M	—	320K	—	11.7	87.5	19.0	70.4	22.8	69.4	10.9	88.2	37.6	63.1
MiDaS [60]	1	ImageNet	1M	—	2M	—	11.1	88.5	23.6	63.0	18.4	75.2	12.1	84.6	33.2	71.5
LeReS [109]	1	ImageNet	1M	—	300K	54K	9.0	91.6	14.9	78.4	17.1	77.7	9.1	91.7	27.1	76.6
HDN [110]	1	ImageNet	14M	—	300K	—	6.9	94.8	11.5	86.7	12.1	83.3	8.0	93.9	<u>24.6</u>	<u>78.0</u>
DPT [61]	1	ImageNet	14M	—	1.2M	188K	9.8	90.3	<u>10.0</u>	90.1	7.8	94.6	8.2	93.4	<b>18.2</b>	75.8
Marigold v1.0 <sup>3</sup> w/ TAESD [27]	1 $\times$ 1	SD v2.0	2.3B	—	—	74K	5.9	96.0	12.2	86.2	7.5	94.4	6.7	95.0	31.7	75.6
Marigold v1.0 LCM	10 $\times$ 1	SD v2.0	2.3B	—	—	74K	5.8	96.1	10.1	<u>90.9</u>	<u>6.6</u>	<u>95.8</u>	6.6	95.0	30.5	77.2
Marigold v1.0 <sup>3</sup>	1 $\times$ 1	SD v2.0	2.3B	—	—	74K	<u>5.7</u>	<u>96.2</u>	11.0	89.1	6.9	95.5	6.6	95.2	31.2	76.6
Marigold v1.0 <sup>3</sup>	10 $\times$ 1	SD v2.0	2.3B	—	—	74K	<u>5.7</u>	<u>96.2</u>	10.9	89.2	6.8	95.6	6.5	95.3	31.0	76.7
Marigold v1.0	10 $\times$ 50	SD v2.0	2.3B	—	—	74K	<b>5.5</b>	<b>96.4</b>	<b>9.9</b>	<b>91.6</b>	<b>6.5</b>	<b>96.0</b>	6.4	95.1	30.8	77.3
Marigold v1.1 <sup>3</sup> w/ TAESD [27] (fp16)	1 $\times$ 1	SD v2.0	2.3B	—	—	74K	6.1	95.8	12.4	85.0	7.6	94.3	6.8	95.1	31.1	75.9
Marigold v1.1 <sup>3</sup> (fp16)	1 $\times$ 1	SD v2.0	2.3B	—	—	74K	5.8	96.1	11.0	88.8	7.0	95.5	6.6	95.3	30.4	77.3
Marigold v1.1 <sup>3</sup>	1 $\times$ 1	SD v2.0	2.3B	—	—	74K	5.9	96.1	11.0	88.8	7.0	95.5	6.6	95.3	30.4	77.3
Marigold v1.1 <sup>3</sup>	10 $\times$ 1	SD v2.0	2.3B	—	—	74K	5.8	96.1	10.9	89.0	6.9	95.7	6.5	95.4	30.3	77.3
Marigold v1.1 <sup>3</sup>	1 $\times$ 4	SD v2.0	2.3B	—	—	74K	<u>5.7</u>	<u>96.2</u>	10.8	89.6	7.2	95.3	<u>6.0</u>	<u>96.0</u>	30.1	77.9
Marigold v1.1 <sup>3</sup>	10 $\times$ 4	SD v2.0	2.3B	—	—	74K	<b>5.5</b>	<b>96.4</b>	10.5	90.2	6.9	95.7	<b>5.8</b>	<b>96.3</b>	29.8	<b>78.2</b>

<sup>1</sup> Metrics in the 1<sup>st</sup> section are sourced from the respective papers. Metrics in the 2<sup>nd</sup> section are sourced from Metric3D [55], except the ScanNet benchmark. For ScanNet, Metric3D used a different random split that is not publicly accessible. Therefore, we re-ran baseline methods on our split. We additionally took numbers from Metric3D for HDN [110] on ScanNet benchmark due to unavailable source code.

<sup>2</sup> Privileged information used by methods: Metric3D and Metric3D v2 require camera intrinsics; GeoWizard requires choosing between indoor and outdoor regimes.

<sup>3</sup> These Marigold variants are evaluated using the trailing timestamps setting of the DDIM scheduler.

TABLE II: **Inference time** of Marigold-Depth and other methods on a  $768 \times 768$  image using an RTX 3090 GPU.

Method	Time (sec)
DPT	0.158
DepthAnything v2	0.289
Metric3D v2	0.386
Marigold v1.1 (1 $\times$ 1)	0.568
Marigold v1.1 (1 $\times$ 1) (fp16)	0.274
Marigold v1.1 (1 $\times$ 1) w/ TAESD (fp16)	0.082

**ETH3D** [113] and **DIODE** [114] are two high-resolution datasets, both featuring depth maps derived from LiDAR sensor measurements. For ETH3D, we incorporate all 454 samples with available ground truth depth maps. For DIODE, we use the entire validation split, which encompasses 325 indoor samples and 446 outdoor samples.

**Evaluation protocol.** Following the protocol of affine-invariant depth evaluation [60], we first align the estimated merged prediction  $\mathbf{m}$  to the ground truth  $\mathbf{d}$  with the least squares fitting. This step gives us the metric depth map  $\mathbf{a} = \mathbf{m} \times s + t$  in the same units as the ground truth. Next, we apply two metrics [55], [60], [61], [109] for assessing quality of depth estimation. The first is Absolute Mean Relative Error (AbsRel ↓), calculated as:  $\frac{1}{M} \sum_{i=1}^M |\mathbf{a}_i - \mathbf{d}_i| / \mathbf{d}_i$ , where  $M$  is the total number of pixels. The second, Threshold Accuracy ( $\delta 1 \uparrow$ ), measures the

proportion of pixels satisfying  $\max(\mathbf{a}_i / \mathbf{d}_i, \mathbf{d}_i / \mathbf{a}_i) < 1.25$ .

**Comparison with other methods.** We compare Marigold-Depth to 5 zero-shot baselines in Tab. I. We filtered baselines based on the affordability of the training protocol (at most 2M training samples) and temporal relevance. Marigold-Depth, built upon the Stable Diffusion prior and a small set of synthetic samples, outperforms prior work and remains competitive with methods that rely on larger training sets.

Although trained exclusively on synthetic depth datasets, the model generalizes well to a wide range of real-world scenes. For visual assessment, we present a qualitative comparison and 3D visualizations of surface normals reconstructed from depth in Figs. 3 and 5. Marigold accurately captures both the overall scene layout, such as the spatial relationships between walls and furniture, and fine-grained details, as indicated by the arrows. In particular, the reconstruction of flat surfaces, especially walls, is significantly improved. Additionally, we report an inference speed comparison in Tab. II.

### G. Ablation Studies

We select two zero-shot validation sets for ablation studies: the official training split of NYUv2 [21], consisting of 785 samples, and a randomly selected subset of 800 images from the KITTI Eigen [7] training split.

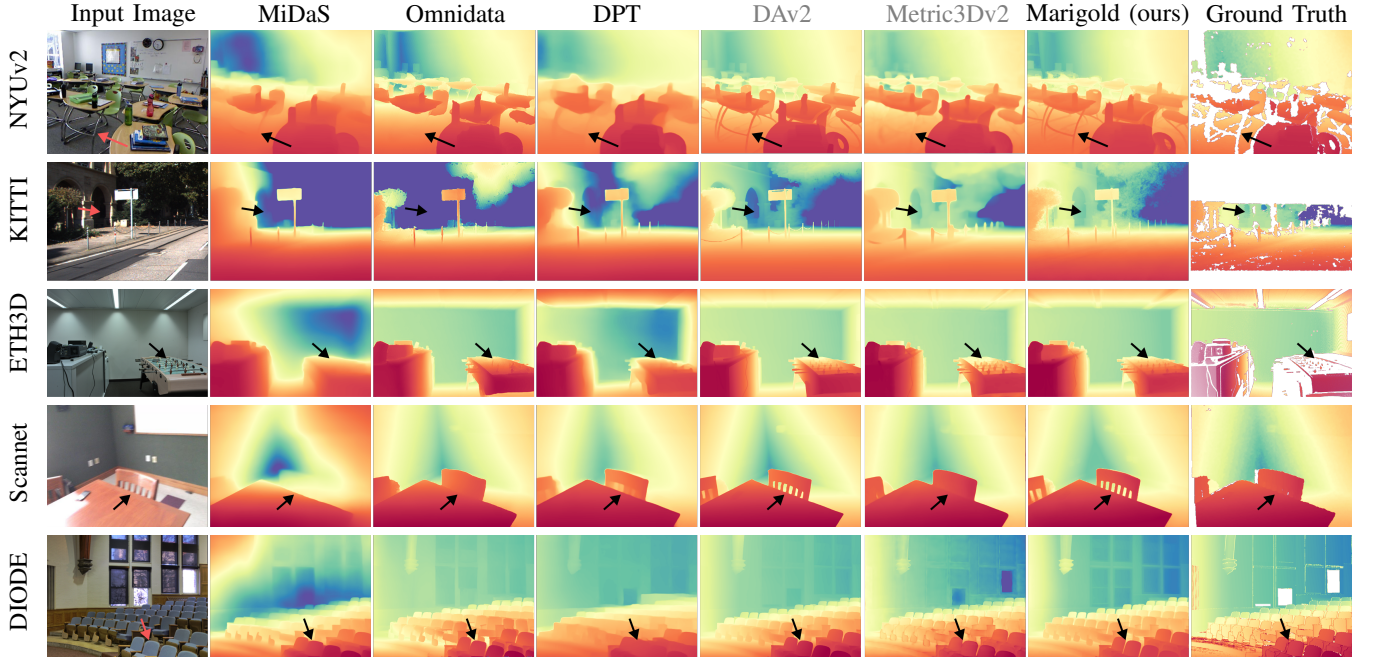


Fig. 3: **Qualitative comparison** of monocular depth estimation methods across different datasets. Marigold excels at capturing thin structures (e.g., chair legs) and preserving overall layout of the scene (e.g., walls in ETH3D and chairs in DIODE).

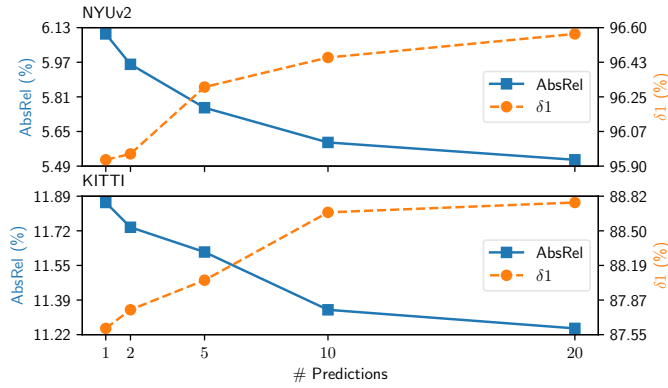


Fig. 4: **Ablation of ensemble size.** We observe a monotonic improvement with the growth of ensemble size. This improvement starts to diminish after 10 predictions per sample.

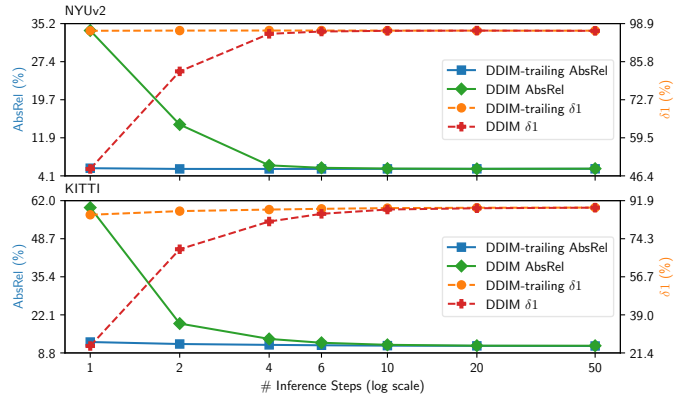


Fig. 6: **Ablation of denoising steps.** One denoising step is sufficient with DDIM-trailing [23] (default in v1.1). DDIM [18] (default in v1.0) requires at least 4-10 steps.

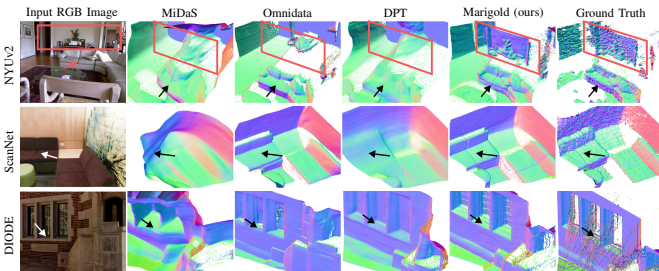


Fig. 5: **Qualitative comparison (unprojected from depth, colored as normals)** of monocular depth estimation methods across different datasets. Marigold-Depth stands out for its superior reconstruction of flat surfaces and detailed structures.

TABLE III: **Training datasets:** HyperSim [66] alone contributes the most; Virtual KITTI [67] improves outdoor results.

HyperSim	Virtual KITTI	NYUv2		KITTI	
		AbsRel ↓	$\delta 1$ ↑	AbsRel ↓	$\delta 1$ ↑
×	✓	13.9	83.4	15.4	79.3
✓	×	5.7	96.3	13.7	82.5
✓	✓	<b>5.6</b>	<b>96.5</b>	<b>11.3</b>	<b>88.7</b>

**Training data domain.** To better understand the impact of the synthetic data on model generalization, we conduct an ablation study using the two datasets employed during training. The results, presented in Tab. III, show that even fine-tuning on a single synthetic dataset enables the pretrained LDM to adapt reasonably well to monocular depth estimation. However,



TABLE IV: **Quantitative Comparison** of Marigold-Normals with SOTA surface normals estimators on several zero-shot benchmarks<sup>1</sup>. In the 1<sup>st</sup> section, we list methods requiring more than 2M samples for fine-tuning. We compare methods from the 2<sup>nd</sup> section with flavors of Marigold-Normals (ours, this paper) from the 3<sup>rd</sup> section. Legend: The mean metric is presented as absolute angles, 11.25° metric is in percentage terms; bold numbers are the best, underscored second best; NFEs is the number of function evaluations required to obtain the prediction – *ensemble* × *steps* for diffusion models and 1 for end-to-end networks. Marigold outperforms other methods in this low-data regime on most indoor and outdoor scenes.

Method	NFEs	Prior	Data			ScanNet		NYUv2		iBims-1		DIODE		OASIS	
			Prior	Real	Synthetic	Mean ↓	11.25° ↑	Mean ↓	11.25° ↑	Mean ↓	11.25° ↑	Mean ↓	11.25° ↑	Mean ↓	11.25° ↑
Metric3D v2 [56] <sup>2</sup>	1	DINOv2	142M	16M	91K	—	—	13.3	66.4	19.6	69.7	12.6	64.9	23.4	28.5
OmniData v2 [81]	1	ImageNet	15M	12M	310K	16.2	60.2	17.2	55.5	18.2	63.9	20.6	40.8	24.2	27.7
DSINE [82] <sup>2</sup>	1	ImageNet	1.3M	86K	74K	16.2	61.0	16.4	59.6	17.1	67.4	19.9	41.8	24.4	28.8
GeoWizard [69] <sup>2</sup>	10 × 50	SD v2.0	2.3B	51K	227K	17.6	54.6	19.0	50.0	19.3	62.3	24.7	30.1	25.3	26.9
GenPercept [26]	1	SD v2.1	2.3B	—	44K	18.2	57.4	18.3	56.0	18.3	63.8	22.3	38.1	25.9	23.3
StableNormal [83]	1	SD v2.0	2.4B	51K	227K	16.7	54.0	17.8	54.2	17.1	67.8	19.3	<b>53.8</b>	25.7	25.4
Lotus-G [25]	1 × 1	SD v2.0	2.3B	—	59K	15.3	64.0	16.9	59.1	17.5	66.1	21.2	39.7	24.7	27.0
Lotus-D [25]	1	SD v2.0	2.3B	—	59K	15.3	62.9	16.8	58.2	17.7	64.9	21.0	39.7	25.7	25.3
E2E-FT [23]	1	Marigold	2.3B	—	74K	14.7	66.0	16.2	<b>61.4</b>	<b>15.8</b>	<b>69.9</b>	19.2	43.8	22.8	29.8
E2E-FT [23]	1	SD v2.0	2.3B	—	74K	14.7	<b>66.1</b>	16.5	60.4	16.1	69.7	19.0	44.4	23.6	27.9
Marigold-Normals v0.1 <sup>3</sup>	10 × 50	SD v2.0	2.3B	—	39K	16.1	62.3	17.1	58.5	16.6	68.0	19.6	44.7	23.5	28.0
Marigold-Normals v1.1	1 × 1	SD v2.0	2.3B	—	77K	14.9	64.3	16.4	58.9	17.2	65.6	19.5	43.2	23.2	28.3
Marigold-Normals v1.1	10 × 1	SD v2.0	2.3B	—	77K	14.8	64.5	16.3	59.0	17.1	65.7	19.5	43.3	23.2	28.3
Marigold-Normals v1.1	1 × 4	SD v2.0	2.3B	—	77K	15.2	65.3	17.0	59.6	17.0	68.0	19.4	45.0	23.2	29.2
Marigold-Normals v1.1	10 × 4	SD v2.0	2.3B	—	77K	<b>14.5</b>	<b>66.1</b>	<b>16.1</b>	<u>60.5</u>	16.3	68.5	<b>18.8</b>	<u>45.5</u>	<b>22.4</b>	<b>30.1</b>

<sup>1</sup> Metrics on ScanNet, NYUv2, and iBims-1 are sourced from the respective papers. Metrics that were not reproducible are re-computed by us (for GeoWizard, GenPercept, and StableNormal). We compute the metrics for all methods on OASIS (except for OmniData and DSINE) and DIODE.

<sup>2</sup> Privileged information used by methods: Metric3D v2 and DSINE require camera intrinsics; GeoWizard requires choosing between indoor and outdoor regimes.

<sup>3</sup> The preview models (v0.1), uploaded to the Hugging Face repository shortly after releasing Marigold-Depth (v1.0) [18].

more diverse and photorealistic data yield stronger performance across both indoor and outdoor scenes. Notably, incorporating training data from a different domain enhances performance not only on that domain but also on the original one.

**Test-time ensembling.** We test the effectiveness of the proposed test-time ensembling scheme by varying the number of predictions. As shown in Fig. 4, a single prediction already yields reasonably good results. Ensembling 10 predictions can reduce the absolute relative error on NYUv2 by ~8%, and ensembling 20 predictions brings an improvement of ~9.5%.

**Number of denoising steps.** Like the base model, Marigold is configured to use a 1000-step DDPM schedule during training. In version 1.0, inference used DDIM with leading timesteps, requiring 4 to 10 function evaluations (NFE) to reach peak performance. To report best-case results, we used 50 steps in the initial evaluations. Later, Garcia *et al.* [23] proposed switching to trailing timesteps for inference with DDIM. This change significantly improved efficiency of Marigold, with performance saturating at just 1 DDIM step (NFE=1). This is the default in all Marigold v1.1 models. Table I reports results across model versions, NFE, ensemble sizes, alternative VAEs, and FP16 quantized weights. The effect of varying denoising steps in DDIM scheduler [30] is shown in Fig. 6.

#### IV. SURFACE NORMALS ESTIMATION MODEL

Surface normals and depth estimation are inherently related, as both aim to regress 3D geometry. While the depth of a pixel is predicted as a single positive scalar, the surface normal is represented as a three-dimensional vector on the unit sphere. Real surface normals can hardly be collected outside of a simulation or a controlled environment. Instead, normals have traditionally been derived from depth measurements, which often introduce noise at flat surfaces and unrealistic smoothness at depth discontinuities. Simulated data, however,

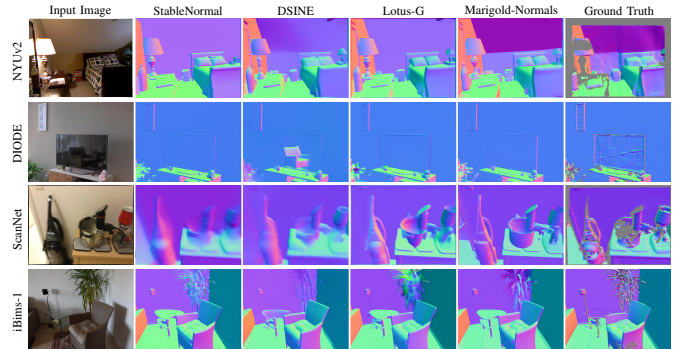


Fig. 7: **Qualitative comparison** of monocular surface normals estimation methods across different datasets. Compared to baseline methods, Marigold-Normals demonstrates superior performance in handling complex scene layouts and shows greater robustness to motion blur and reflections.

often struggles with the sim-to-real gap. This motivates Marigold-Normals, which aims to bridge the gap to real data through its Stable Diffusion prior.

##### A. Method

We closely follow the Marigold-Depth fine-tuning protocol and introduce Marigold-Normals, a model variant for monocular surface normals estimation. Methodology adaptation for the new task to cope with the three-dimensional unit vectors of surface normals is minimal. Raw ground-truth normal maps are streamed directly into the VAE encoder during training. No range normalization or channel replication is required. VAE outputs are normalized along the channel dimension to ensure unit-length predictions.

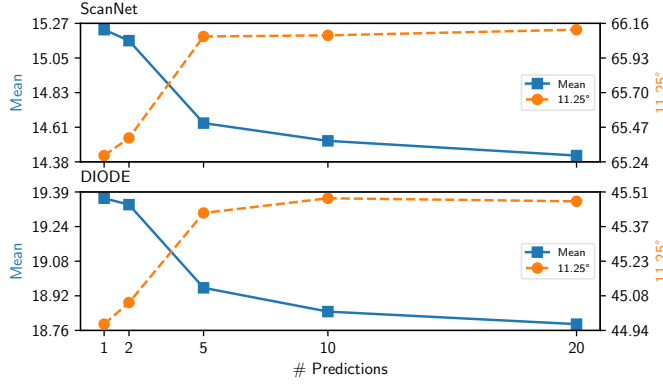


Fig. 8: **Ablation of ensemble size** for Marigold-Normals. The performance consistently improves with increasing ensemble size. Diminishing returns begin after 10 predictions per sample.

**Test-time ensembling.** First, we run inference  $N$  times with different noise initialization. We then average the normals predictions  $\{\hat{\mathbf{n}}_1, \dots, \hat{\mathbf{n}}_N\}$  to a single mean prediction  $\bar{\mathbf{n}}$  and normalize it to unit length. Lastly, for every pixel  $(u, v)$  in the final prediction, we select the nearest neighbor vector from the  $i$ -th prediction  $\hat{\mathbf{n}}_i^{(u,v)}$  that maximizes the cosine similarity with the corresponding vector in the mean normal map  $\bar{\mathbf{n}}^{(u,v)}$ :  $\arg \max_{i \in \{1, \dots, N\}} \bar{\mathbf{n}}^{(u,v)} \cdot \hat{\mathbf{n}}_i^{(u,v)}$ .

### B. Implementation

We fine-tune the model for 26K iterations using the Adam optimizer with a learning rate of  $6 \cdot 10^{-5}$ . The training data is augmented with horizontal flipping, blurring, and color jitter. At inference, we use the DDIM scheduler [30] in trailing mode and perform 4 steps. The final prediction is aggregated using an ensemble size of 10. All other settings are the same as for the depth model.

**Training Datasets.** We train the model on three synthetic datasets covering both indoor and outdoor scenes. HyperSim [66] and InteriorVerse [99] are photo-realistic indoor datasets. We filter out incomplete samples and obtain 49K samples from 434 scenes for HyperSim and 27K samples from 3806 scenes for InteriorVerse. The training resolution is kept at  $480 \times 640$  for both datasets. Sintel [115] consists of indoor and outdoor sequences from a short animated film. We filter out low-quality samples and use 627 training images. The images are center-cropped to an aspect ratio of 3 : 4 and resized to  $480 \times 640$ . Despite the small sample size, we observe improvement in training with Sintel due to its scene diversity and image appearance.

### C. Evaluation

**Evaluation Protocol.** We evaluate our method on five unseen benchmarks. NYUv2 [21], ScanNet [112], and iBims-1 [116] are indoor depth datasets, for which we use the ground-truth normals provided by [82]. This includes 654 samples from the NYUv2 test split, all 100 samples of iBims-1, and a subset of 300 samples defined by [78] of ScanNet. DIODE [114] features both indoor and outdoor scenes. We use the validation

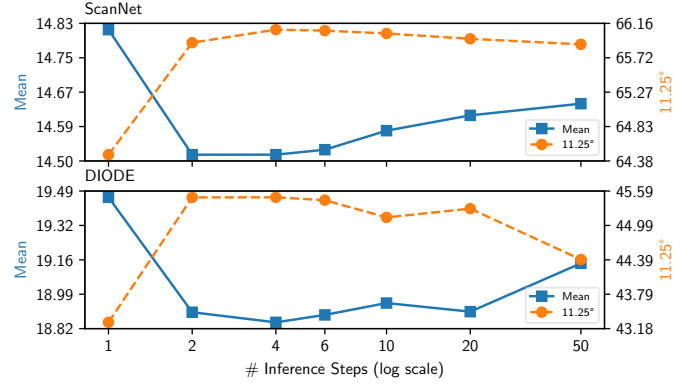


Fig. 9: **Ablation of denoising steps** for Marigold-Normals. The best performance in benchmarks is obtained at 4 denoising steps. Visually, one step is sufficient in most cases.

split, encompassing 325 indoor and 446 outdoor samples. OASIS [117] contains in-the-wild images sourced from the internet. We use the entire validation split of 10,000 samples. Following established methods [78], [118], [119], we report the mean angular error and the percentage of pixels with an angular error  $< 11.25^\circ$ .

**Comparison with other methods.** We compare Marigold-Normals to 8 baselines. DSINE [82] is a discriminative regression-based method that relies on a CNN architecture. Similar to us, GeoWizard [69], StableNormal [83], and Lotus-G [25] are generative diffusion-based methods that fine-tune a Stable Diffusion backbone. GenPercept [26], Lotus-D [25], and E2E-FT [23] bypass the probabilistic formulation and perform end-to-end training and inference instead.

The quantitative results are shown in Table IV. Our method consistently outperforms the baselines on most datasets and metrics. Notably, our straightforward fine-tuning and inference approach proves more effective than the more complex strategies employed by other diffusion-based methods. A qualitative comparison is shown in Figure 7. It is apparent that Marigold-Normals produces highly accurate predictions, even in challenging scene layouts and scenarios.

### D. Ablations

**Test-time ensembling.** We investigate the impact of test-time ensembling while varying the number of aggregated predictions. The ablation studies are conducted on the ScanNet and DIODE splits. As illustrated in Fig. 8, the performance with ensembling behaves very similarly to the depth model. While a single prediction already delivers solid results, the performance consistently improves as more predictions are combined. However, like depth, the gains plateau when more than 10 predictions per image are aggregated.

**Number of denoising steps.** As quantitatively shown in Fig. 9, the best performance with the DDIM trailing timesteps setting is achieved at 4 steps. We additionally visualized predictions with 1-, 4-, and 20-step inference in Fig. 10. Interestingly, the level of detail can be controlled by simply adjusting the number of denoising iterations. While 1-step inference already

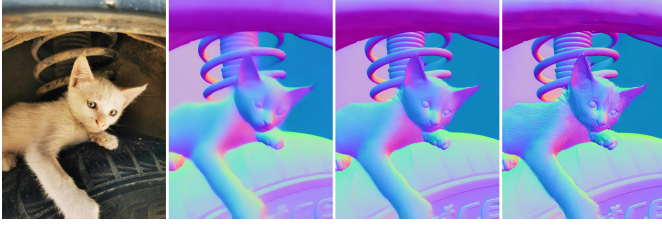


Fig. 10: **Prediction granularity and denoising steps.** From left to right, we visualize predictions with 1, 4, and 20 denoising steps during inference. By increasing the number of steps, fine details, such as the cat’s fur, become more pronounced.

produces reasonably good results in all cases, increasing the number of steps results in more pronounced details in high-frequency regions. However, improved details do not necessarily translate to improved performance metrics, as most evaluation benchmarks either mask out or over-smooth high-frequency regions in the ground truth.

## V. INTRINSIC DECOMPOSITION MODELS

Adaptation of Marigold to other image analysis tasks, such as Intrinsic Image Decomposition (IID), is simple and affordable. Specifically, this task enables a structured separation of images into physically meaningful properties. We introduce two models: Marigold-IID-Appearance and Marigold-IID-Lighting.

Marigold-IID-Appearance represents intrinsic properties through a physically-based BRDF, where material attributes are characterized by three key components: albedo, roughness, and metallicity. This model primarily focuses on estimating illumination-independent reflectance properties.

Marigold-IID-Lighting decomposes an image into albedo, diffuse shading, and a non-diffuse residual component. This decomposition aligns with the intrinsic residual model in linear space  $I = A \cdot S + R$ , where the image  $I$  is composed of albedo  $A$ , a diffuse shading component  $S$  (representing illumination color), and an additive residual term  $R$  capturing non-diffuse effects. This formulation provides a structured way to separate reflectance from shading while accounting for complex illumination phenomena.

Similarly to the surface normals estimation task, ground truth for IID is hard to obtain without simulation or outside a controlled capture environment. Therefore, we again turn to the available synthetic data to derive the Marigold-IID models.

### A. Method

Different from depth and normals, IID requires predicting multiple images representing the decomposition of a single input image. In the case of Marigold-IID-Appearance, we predict two images per input: the first is the albedo image in standard color space, with values normalized to the unit range. The second image encodes two BRDF properties (roughness and metallicity) into the red and green channels [96], also normalized to a unit range. We denote this modality as *material*. We keep the blue channel at zero, which gives the material visualization a red-green tone. For Marigold-IID-Lighting, three images are predicted per input: the albedo image, shading, and residual components, all normalized to the unit range.

TABLE V: **Quantitative Comparison** of Marigold-IID-Appearance v1.1 with SOTA methods on the InteriorVerse test set. Marigold outperforms the two competing methods on this benchmark.

Method	Albedo			Material		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
IID-Diffusion [96]	18.10	<b>0.863</b>	0.198	16.09	0.691	0.356
RGB $\leftrightarrow$ X [97]	13.16	0.774	0.289	10.13	0.547	0.636
Marigold-IID-Appearance v1.1	<b>19.50</b>	<u>0.846</u>	<b>0.190</b>	<b>17.63</b>	<b>0.803</b>	<b>0.286</b>

TABLE VI: **Quantitative Comparison** of Marigold-IID-Lighting v1.1 with SOTA methods on the HyperSim test set. Marigold achieves competitive performance on this benchmark.

Method	Albedo			Lighting		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
IID-in-the-wild [90]	<b>19.28</b>	<b>0.819</b>	0.260	<u>16.82</u>	0.725	0.308
RGB $\leftrightarrow$ X [97]	17.43	<u>0.795</u>	<b>0.200</b>	16.70	<b>0.742</b>	<b>0.251</b>
Marigold-IID-Lighting v1.1	<u>18.21</u>	0.771	<u>0.218</u>	<b>17.62</b>	<u>0.729</u>	<u>0.263</u>

### B. Implementation

The U-Net is adapted to handle the increased number of predicted images ( $P$ ): 2 and 3 for the IID-Appearance and IID-Lighting models, respectively. The input channels of the first convolutional layer are replicated  $P + 1$  times; the whole weight tensor is divided by the replication factor to maintain the activations statistics. The output channels of the final layer are replicated  $P$  times without changing weights.

The modified IID-Appearance U-Net is fine-tuned for 40K iterations on the training split of InteriorVerse, consisting of 45K samples at  $480 \times 640$  resolution. Gamma correction and conversion from linear to sRGB space are applied to the input scene and target intrinsic images. The IID-Lighting U-Net is fine-tuned for 36K iterations on a pre-filtered training split of HyperSim, yielding 24K samples. All samples are resized to  $480 \times 640$  and converted to sRGB space while keeping the target intrinsic images in linear space.

For quantitative evaluation, we perform 4 denoising steps without ensembling. Other settings remain the same as for the depth model. As with other Marigold models, 1 step is sufficient qualitatively.

### C. Evaluation

**Evaluation Protocol.** The IID-Appearance model is evaluated on the test split of the InteriorVerse [99] dataset, which contains 2.6K samples. We assess the prediction performance of albedo and material. For IID-Lighting, evaluation is performed on the HyperSim [66] test split, comprising 5.2K samples, where we evaluate the quality of the predicted albedo and shading components. Albedo and material predictions are compared directly to the ground truth without any alignment. Due to their differing value ranges, shading predictions are first scale-aligned to the corresponding ground truth and then normalized to the unit range before evaluation. The reported metrics are Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [120], and Learned Perceptual Image Patch Similarity (LPIPS) [121].



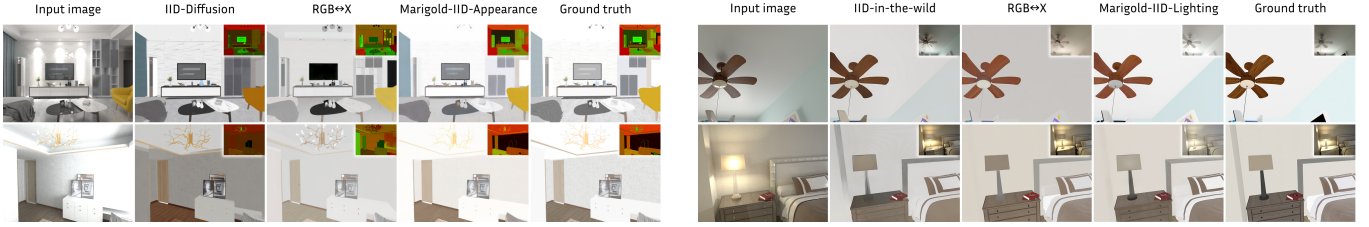


Fig. 11: **Qualitative comparison** of Marigold-IID-Appearance (left) and Marigold-IID-Lighting (right). **Left:** albedo and material (with roughness in the red channel and metallicity in the green channel) predictions on the InteriorVerse test set. **Right:** albedo and diffuse shading predictions on the HyperSim test set. Predictions of Marigold-IID-Appearance and Marigold-IID-Lighting contain less baked-in shading and are more consistent with the ground truth.

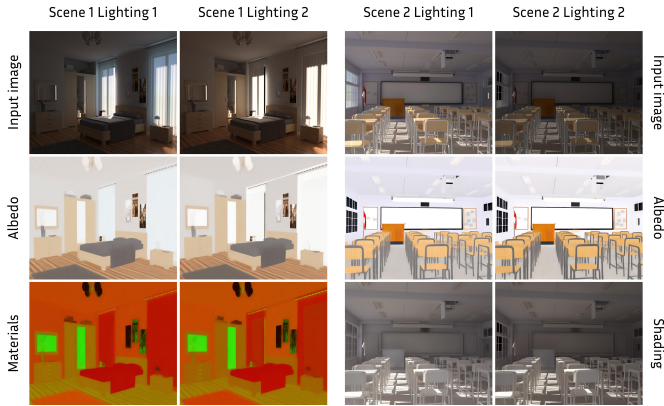


Fig. 12: **Robustness to varying lighting conditions.** The Marigold-IID models generate consistent predictions across different environmental lighting setups of the same scene.

**Comparison with other methods** We compare our results to three state-of-the-art methods: RGB↔X [97] and Intrinsic Image Diffusion (IID-Diffusion) [96] for IID-Appearance, and RGB↔X and IID-in-the-wild [90] for IID-Lighting. For the cited methods, we adopt the inference settings reported in their respective papers: 50 denoising steps for RGB↔X, and 50 denoising steps with an ensemble size of 10 for IID-Diffusion.

The quantitative comparisons are shown in Tab. V and Tab. VI. The visual comparisons are presented in Fig. 11. Our method produces quantitatively more accurate and qualitatively cleaner decompositions of images. We further demonstrate the robustness of our method to varying environmental lighting conditions in Fig. 12.

**All modalities in-the-wild** can be seen in Fig. 13. Evidently, image analysis tasks benefit from the rich generative prior, validating our universal fine-tuning protocol.

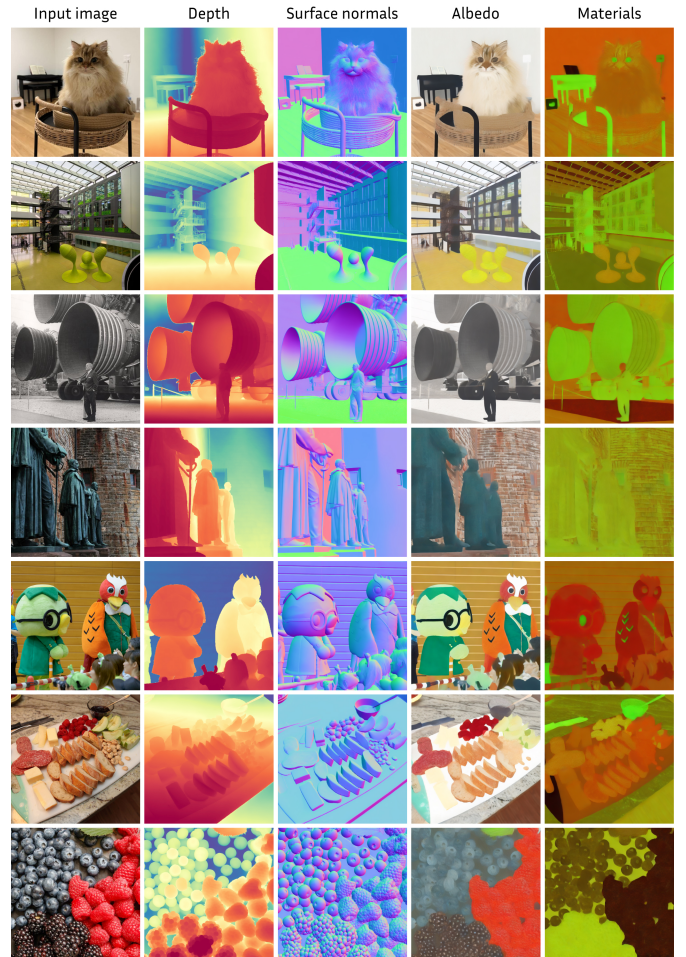


Fig. 13: **Marigold in-the-wild results – all modalities.** Our fine-tuning protocol enables generalization across multiple modalities. None of the fine-tuning datasets included humans, animals, food, engines, or toys, attesting to the successful carry-over of the rich generative prior to downstream tasks.

## VI. LATENT CONSISTENCY MODEL (LCM)

Latent Consistency Models [22] is a latent diffusion model class that enables high-quality one- or few-step inference. Inspired by LCM’s success in fast image generation, we developed Marigold-LCM, a Latent Consistency Model variant of Marigold that achieves similar prediction results in one or a few denoising steps.

### A. Method

We distill Marigold-LCM from the standard Marigold using a similar recipe detailed in Luo *et al.* [22] (Fig. 14). Similarly to the base fine-tuning protocol, we only distill the U-Net part of the model while keeping the VAE frozen. The LCM distillation involves three models: a frozen *teacher* model  $\Phi$ , which is a standard Marigold U-Net; a *student* model  $\Theta$ , which

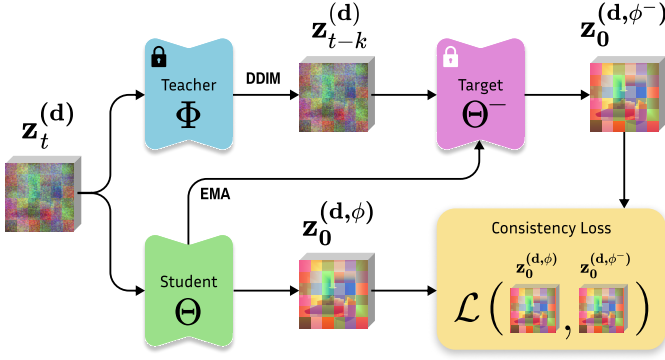


Fig. 14: **Overview of Marigold-LCM distillation.** To train Marigold-LCM, we initialize three replicas of the base Marigold: the Teacher ( $\Phi$ ), the Target ( $\Theta^-$ ), and the Student ( $\Theta$ ). The student is updated via optimization of the consistency objective, and the target is updated via EMA of student weights. At each training step, the student learns to predict the same clean latent  $\mathbf{z}_0^{(d)}$  as produced by the teacher after applying the DDIM step of size  $k$ . Each model takes the image condition  $\mathbf{z}^{(x)}$  and the input timestep, similarly to Fig. 1.

we eventually output as Marigold-LCM; and a *target* model  $\Theta^-$ . Both the *student* and *target* models are initialized with the same weights as the teacher model.

At each training iteration, we sample data from the training dataset and convert the sample to latent images using the VAE encoder. We then sample Gaussian noise and diffuse the depth latent to a random timestep  $t$  to obtain  $\mathbf{z}_t^{(d)}$ . Now the *teacher* model takes  $\mathbf{z}_t^{(d)}$  as input, predicts noise, and then estimates the depth latent  $\mathbf{z}_{t-k}^{(d)}$  at noise level  $t-k$  using the DDIM solver step detailed in [22]. We set  $k = 200$  in our implementation, which yields the best distillation result. We now minimize the loss between the outputs of *student*  $\Theta$  and *target* model  $\Theta^-$  through a self-consistency function  $\mathbf{f}$  [22], [31]

$$\mathcal{L}(\mathbf{f}(\Theta, \mathbf{z}_t^{(d)}, t), \mathbf{f}(\Theta^-, \mathbf{z}_{t-k}^{(d)}, t-k)). \quad (5)$$

Here, the self-consistency function is defined as

$$\mathbf{f}(\theta, \mathbf{x}, t) = c_{\text{skip}}(t)\mathbf{x} + c_{\text{out}}(t)\mathbf{z}_0^{(d, \theta)}, \quad (6)$$

where  $c_{\text{skip}}, c_{\text{out}}$  are differentiable functions that satisfy  $c_{\text{skip}}(\epsilon) = 1, c_{\text{out}} = 0$  for some small  $\epsilon > 0$ , and  $\mathbf{z}_0^{(d, \theta)}$  is the clean denoised depth latent predicted by model  $\theta$ . We use the Pseudo-Huber metric [122] as our loss function:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = \sqrt{\|\mathbf{x} - \mathbf{y}\|_2^2 + c^2} - c, \quad (7)$$

where we set  $c = 0.001$ .

At the end of each training iteration, the *student* model is updated using gradient descent according to our loss function. While the weights of the *target* model are updated with a running average of the weights of the *student* model:

$$\Theta^- = \text{stopgrad}(\mu\Theta^- + (1 - \mu)\Theta), \quad (8)$$

here the decay weight  $\mu$  is set to be 0.95.

## B. Inference

The consistency distillation training allows only one forward pass of the noise latent through the trained student model to generate the clean predicted depth latent. Additionally, one can improve the sample quality by taking multi-step sampling [22], [31]. This is done by adding another random noise latent to the denoised latent following a timestep schedule and then denoising again through the student model. In practice, we use the identical noise schedule of the original Marigold, swap the U-Net with the trained LCM model, and change the denoising sampling method from DDIM to LCM, as introduced above. This means that the evaluation protocol of Marigold-LCM is identical to the base Marigold.

## C. Implementation

The Marigold-LCM model is distilled from the base model for 5K iterations using the AdamW [123] optimizer with a base learning rate of  $3 \cdot 10^{-6}$ . We use the same training set, batch size, gradient accumulation steps, and data augmentation as Marigold training. Distilling Marigold LCM takes approximately one day on a single NVIDIA A100 GPU card with 40G VRAM. We apply one-step denoising at inference time to output the clean predicted depth latent. For evaluation, we follow the same ensemble method as the standard Marigold using 10 samples.

## D. Experiments

We compared Marigold-LCM with one LCM inference step with various Marigold DDIM configurations in Tab. I. Although Marigold-LCM with one step does not outperform the original Marigold with 50 steps in most cases, it outperforms prior art on most datasets and metrics. This validates the hypothesis that Marigold is amenable to the latent consistency distillation, and the resulting model is on par with the base. It also shows that LCM distillation can be successfully adapted to modalities other than text-to-image. However, given the improved quantitative and qualitative performance of Marigold with DDIM and trailing timesteps pointed out in E2E-FT [23], the viability of LCM distillation remains an open question for future research.

## VII. HIGH RESOLUTION DEPTH MODEL

Applying monocular depth estimation networks to high-resolution images seems straightforward, but poses two inherent challenges. First, neural networks have a fixed receptive field limited to the model architecture or the resolution of the training data. Second, memory consumption can become excessive when naively applying those models to larger image dimensions. The simple way to address those challenges is to downsample to its native processing resolution (768 for Marigolds fine-tuned from Stable Diffusion) and later upsample the computed depth map. We refer to this prediction as a “global depth map”  $\hat{\mathbf{d}}^{(g)}$ . Typically, this approach compromises the edge quality of the depth estimation for high-resolution predictions. An alternative approach is partitioning a large image into smaller patches and processing them independently. However, even if consistency at the seams was perfect, this method suffers from global inconsistencies due to the lack of communication between neighboring patches w.r.t. global layout.



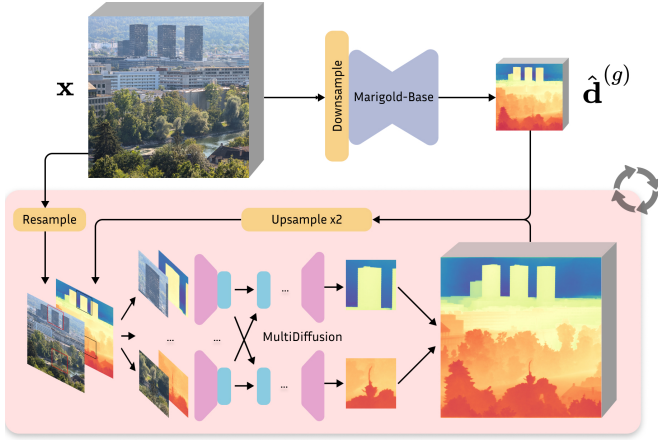


Fig. 15: **High-resolution Marigold Pipeline.** We first create a global prediction  $\hat{d}^{(g)}$  with the original Marigold-Depth pipeline at the native processing resolution. This prediction is then used as an additional conditioning variable in the upsampling diffusion process, which upsamples the prediction in a patch-based, MultiDiffusion forward pass.

#### A. Method

We introduce the Marigold-HR model to overcome these challenges (Fig. 15). The process starts by predicting a global depth map at native processing resolution, as in the original version. This global depth map serves as a coarse scene representation and is the initialization for the following refinement procedure.

Next, we upsample the global depth map by a factor  $2\times$  and use it with the correspondingly resampled RGB image as the conditioning for another diffusion model named  $\Phi$ . To keep memory usage bounded, we implement the forward pass as a bundle of inferences of overlapping tiles, where we synchronize latents via a closed-form equation from the MultiDiffusion approach [103]:

$$\Psi(J_t | z) = \sum_{i=1}^n \frac{F_i^{-1}(W_i)}{\sum_{j=1}^n F_j^{-1}(W_j)} \otimes F_i^{-1}(\Phi(z_t^i)) \quad (9)$$

$W_i$  are the per-pixel blending weights of tile  $i$  – in our implementation, we use the Chamfer distance to the image border. The function  $F_i^{-1}$  transforms the tile back into its location in the global canvas based on the tile index  $i$ . For this model latent variable  $z_t^i$  is defined as the  $F_i(\text{cat}(z_t^{(d)}, z_t^{(g)}, z^{(x)}))$ .

#### B. Implementation

For the Marigold-HR refiner, we resume the training from the Marigold-Depth checkpoint for 12K iterations with additional conditioning on the  $\times 2$  lower resolution inference. We follow the BetterDepth protocol [70], first aligning the global prediction to the ground truth and then masking out the loss of dissimilar patches, such that the refiner is encouraged to follow the conditioning. We use the suggested setting of  $\eta = 0.1$  to threshold the distance between patches. For the MultiDiffusion pipeline, we set the patch overlap to 50%.

We keep the same training datasets as the base model. However, we deviate from the base protocol and train with

half-resolution crops of size  $384 \times 512$ . To generate the global conditioning, we precompute Marigold-Depth results for the whole training dataset with half the dataset resolution as the processing resolution. Specifically, we generate two sets of predictions: one set of base predictions with 10 ensemble members and one set using ensemble size one. We randomly select one of these sets for each sample during training. For data augmentations, we apply Gaussian blur to 50% training samples with random radii ranging from 0 to 4 pixels.

#### C. Evaluation

**Evaluation datasets.** We benchmark on two high-resolution datasets that contain unmasked depth continuities. The datasets are: First, the stereo-matched Middlebury 2014 [124] contains ground truth of resolution  $2016 \times 2940$ . We evaluate the complete dataset with 46 samples. Second, we evaluate the Booster dataset [125], which contains a stereo-generated ground-truth of size  $3008 \times 4112$ . We only consider the scenes where the ground truth is provided. Since the images in each scene only vary by illumination and small parallax, we use one image per scene – 31 samples total.

**Evaluation protocol.** We evaluate all depth maps with the affine-invariant protocol and provide the corresponding general-purpose metrics – i.e., the Absolute Mean Relative Error (AbsRel  $\downarrow$ ) and Threshold Accuracy ( $\delta 1$   $\uparrow$ ) accuracy. Furthermore, four edge-based metrics evaluate the quality of the discontinuities: Depth Boundary Error Completeness ( $\epsilon_{\text{DBE}}^{\text{comp}}$   $\downarrow$ ) and Accuracy ( $\epsilon_{\text{DBE}}^{\text{acc}}$   $\downarrow$ ) [126], as well as Edge Precision ( $\epsilon^{\text{prc}}$   $\uparrow$ ) and Recall ( $\epsilon^{\text{rec}}$   $\uparrow$ ) [127]. Since none of the methods directly outputs the exact dataset-specific resolutions, we resample the prediction using bilinear interpolation. For ours, we employ two upsampling iterations with Marigold-HR, until the output resolution approximates the target resolution.

**Comparison with other methods.** We benchmark our method against other recent methods designed for high-resolution inference. BoostingDepth [100] implements a version based on MiDaS [60] and another based on LeRes [109]. We note that the officially provided checkpoints are trained on a mixed dataset, including our evaluation dataset, Middlebury 2014. Finally, we also compare the recent PatchFusion [101] that bootstraps ZoeDepth [54] and the concurrently proposed DepthPro [104].

In Table VII, we present the results on the high-resolution dataset. Marigold-HR achieves the best or second-best performance in all metrics. Depth Pro performs best in global depth estimation metrics (AbsRel and  $\delta 1$ ). In terms of edge quality metrics ( $\epsilon_{\text{DBE}}^{\text{comp}}$ ,  $\epsilon_{\text{DBE}}^{\text{acc}}$ ,  $\epsilon^{\text{prc}}$ ,  $\epsilon^{\text{rec}}$ ), Depth Pro is also a strong performer; however, on the Booster dataset, Marigold-HR achieves slightly better performance.

Furthermore, we show qualitative results in Fig. 16. These results demonstrate the visual quality attainable with a diffusion-based model at high resolution. For the in-the-wild example as well, our model produces plausible results, capturing even fine-grained details such as the cat’s whiskers.

#### D. Ablations

We conduct ablation studies to evaluate the impact of our model’s two main methodological design choices: global



TABLE VII: **Quantitative comparison** of Marigold-HR v1.0 against SOTA depth estimators. We note the bootstrapped model in the brackets after the actual method name. Marigold-HR improves upon the base model, particularly excelling in edge quality metrics, where it achieves results competitive with current state-of-the-art models.

Method	Data		Middlebury 2014								Booster Dataset					
	Real	Synthetic	AbsRel ↓	$\delta 1$ ↑	$\epsilon_{DBE}^{comp}$ ↓	$\epsilon_{DBE}^{acc}$ ↓	$\epsilon_{prc}^{prc}$ ↑	$\epsilon_{rec}^{rec}$ ↑	AbsRel ↓	$\delta 1$ ↑	$\epsilon_{DBE}^{comp}$ ↓	$\epsilon_{DBE}^{acc}$ ↓	$\epsilon_{prc}^{prc}$ ↑	$\epsilon_{rec}^{rec}$ ↑		
Boosting [100] (MiDaS [60])	2M	—	7.3	94.2	5.9	1.9	30.	21.	7.5	95.2	8.1	2.6	43.	12.		
Boosting [100] (LeRes [109])	300K	54K	7.0	95.3	5.4	1.9	29.	29.	6.9	94.7	6.2	2.2	44.	25.		
PatchFusion [101] (ZoeDepth [54])	3.9M	1M	6.4	96.2	4.5	1.9	29.	42.	6.7	94.6	5.7	2.4	44.	28.		
Depth Pro [104]	5.1M	2.5M	<b>3.1</b>	<b>99.3</b>	<b>3.6</b>	<b>1.7</b>	<b>36.</b>	<b>54.</b>	<b>2.0</b>	<b>99.8</b>	<b>4.9</b>	2.3	<b>48.</b>	<b>38.</b>		
Marigold-Depth v1.0	—	74K	<u>5.0</u>	97.5	5.0	2.4	24.	29.	<b>3.9</b>	99.1	6.4	3.1	33.	22.		
Marigold-HR v1.0	—	74K	<u>5.0</u>	<u>97.8</u>	<b>3.5</b>	<u>1.8</u>	<u>33.</u>	<b>61.</b>	<u>4.3</u>	<u>99.2</u>	<b>4.1</b>	<b>1.8</b>	<b>48.</b>	<b>45.</b>		

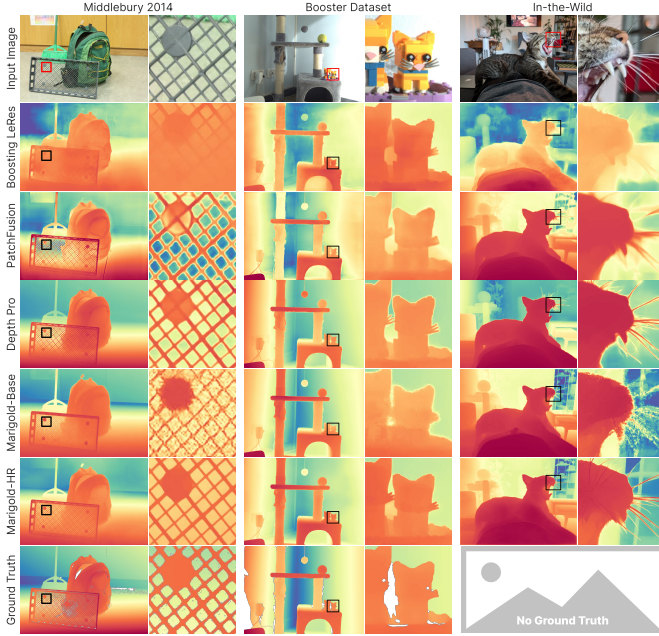


Fig. 16: **Quantitative comparison** of Marigold-HR, Marigold, and other SOTA methods. Predictions on Middlebury 2014 and Booster are aligned to the ground truth and visualized with the same color mapping. The predictions for the in-the-wild example are per sample normalized. Marigold-HR produces fine-grained outputs while also maintaining global context.

conditioning and MultiDiffusion inference. The results are presented in Table VIII.

Starting with the base Marigold-Depth model, we first augment it with global conditioning, which requires retraining the model. This modification improves the edge quality metrics, while the global metrics only slightly worsen. However, we note that GPU memory consumption scales quadratically with the upsampling factor in this configuration, making it less practical for high-resolution applications. Next, we apply the MultiDiffusion inference strategy to the Marigold-Depth model without retraining. This approach also improves the edge quality metrics at higher resolutions but keeps GPU memory bounded. However, we observe a degradation in the global metrics because the patches processed during inference lack global context. Finally, by combining global conditioning and MultiDiffusion inference, we improve global and edge-focused metrics as shown in the last two rows of Table VIII; effectively

TABLE VIII: **Ablation study for Marigold’s high-resolution module.** Combining both global conditioning and MultiDiffusion inference yields the best overall performance compared to its ablated versions.

Method	Size	Middlebury 2014					Booster Dataset				
		$\delta 1$ ↑	$\epsilon_{DBE}^{comp}$ ↓	$\epsilon_{DBE}^{acc}$ ↓	$\epsilon_{prc}^{prc}$ ↑	$\epsilon_{rec}^{rec}$ ↑	$\delta 1$ ↑	$\epsilon_{DBE}^{comp}$ ↓	$\epsilon_{DBE}^{acc}$ ↓	$\epsilon_{prc}^{prc}$ ↑	$\epsilon_{rec}^{rec}$ ↑
Marigold-Depth (base)	768	97.5	5.0	2.4	24.	29.	99.1	6.4	3.1	33.	
	1536	95.3	4.4	2.3	23.	23.	92.6	5.6	2.5	37.	
	3072 <sup>‡</sup>	84.1	6.3	2.2	23.	23.	79.3	8.8	7.0	16.	
Marigold-Depth + Global cond.	1536	97.4	<u>3.7</u>	1.9	<u>30.</u>	30.	98.4	4.5	2.1	45.	
	3072 <sup>‡</sup>	96.7	4.0	<b>1.8</b>	30.	30.	98.1	<u>4.4</u>	1.9	<b>49.</b>	
Marigold-Depth + MultiDiffusion	1536	93.3	4.0	2.1	25.	25.	88.6	4.5	<b>1.6</b>	40.	
	3072	82.9	5.0	2.0	23.	23.	82.9	5.0	2.0	23.	
Marigold-HR (best)	1536	<b>98.0</b>	3.8	2.0	<u>30.</u>	30.	98.8	4.5	2.2	43.	
	3072	<u>97.8</u>	<b>3.5</b>	<b>1.8</b>	<b>33.</b>	<b>33.</b>	<b>99.2</b>	<b>4.1</b>	<u>1.8</u>	<u>48.</u>	

<sup>‡</sup> denotes that inference had to be done in float16 to reduce GPU memory usage.

balancing the benefits of global context and high-resolution edges, with moderate memory costs below 15GB.

## VIII. CONCLUSION

We have presented Marigold, an affordable fine-tuning protocol for pretrained text-to-image LDMs, and a family of models for state-of-the-art image analysis tasks. Our evaluation confirms the value of leveraging rich scene priors and diverse synthetic data across tasks such as monocular depth prediction, surface normals estimation, and intrinsic image decomposition. Marigold offers competitive performance across all these tasks. Additionally, we have presented LCM distillation and High-Resolution inference, which are adaptable to any modality. Marigold is trainable in under 3 GPU-days on consumer hardware, and its single-step inference has runtime comparable with other recent approaches. List of models, web apps (spaces), code, and further reading links:

Space Depth	<a href="https://hf.co/spaces/prs-eth/marigold">hf.co/spaces/prs-eth/marigold</a>
Space Normals	<a href="https://hf.co/spaces/prs-eth/marigold-normals">hf.co/spaces/prs-eth/marigold-normals</a>
Space Intrinsic	<a href="https://hf.co/spaces/prs-eth/marigold-intrinsic">hf.co/spaces/prs-eth/marigold-intrinsic</a>
Model Depth v1.0	<a href="https://hf.co/prs-eth/marigold-depth-v1-0">hf.co/prs-eth/marigold-depth-v1-0</a>
Model Depth v1.1	<a href="https://hf.co/prs-eth/marigold-depth-v1-1">hf.co/prs-eth/marigold-depth-v1-1</a>
Model Normals v1.1	<a href="https://hf.co/prs-eth/marigold-normals-v1-1">hf.co/prs-eth/marigold-normals-v1-1</a>
Model Appearance v1.1	<a href="https://hf.co/prs-eth/marigold-iid-appearance-v1-1">hf.co/prs-eth/marigold-iid-appearance-v1-1</a>
Model Lighting v1.1	<a href="https://hf.co/prs-eth/marigold-iid-lighting-v1-1">hf.co/prs-eth/marigold-iid-lighting-v1-1</a>
Model Depth-LCM v1.0	<a href="https://hf.co/prs-eth/marigold-depth-lcm-v1-0">hf.co/prs-eth/marigold-depth-lcm-v1-0</a>
Model Depth-HR v1.0	<a href="https://hf.co/prs-eth/marigold-depth-hr-v1-0">hf.co/prs-eth/marigold-depth-hr-v1-0</a>
Training code	<a href="https://github.com/prs-eth/Marigold">github.com/prs-eth/Marigold</a>
Inference code	<a href="https://hf.co/docs/diffusers/api/pipelines/marigold">hf.co/docs/diffusers/api/pipelines/marigold</a>
diffusers tutorial	<a href="https://hf.co/docs/diffusers/using-diffusers/marigold_usage">hf.co/docs/diffusers/using-diffusers/marigold_usage</a>

## ACKNOWLEDGEMENTS

We thank: The Hugging Face team (and especially Ahsen Khaliq, Omar Sanseviero, Sayak Paul, Yiyi Xu) for (1) the GPU grants to host Marigold spaces and models, (2) helping us to promote Marigold among research and content creator communities on X (Twitter), Posts, Spaces, and beyond, (3) guidance with integrating Marigold into the diffusers [14]; Robert Presl for creating multiple promotional 3D prints for this paper; Alexander Becker, Dominik Narnhofer, and Xiang Zhang for discussions related to Marigold-HR; Peter Kocsis for help with reproducing [96].

## REFERENCES

- [1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *CVPR*, 2009. 1
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *NeurIPS*, 2012. 1
- [3] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *ICLR*, 2015. 1
- [4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016. 1
- [5] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” *NeurIPS*, 2014. 1
- [6] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015. 1, 2
- [7] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *NeurIPS*, 2014. 1, 3, 6, 7
- [8] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., “Pytorch: An imperative style, high-performance deep learning library,” *NeurIPS*, 2019. 2
- [9] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, “Zero-shot text-to-image generation,” *OpenAI*, 2021. 2
- [10] Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution,” *NeurIPS*, vol. 32, 2019. 2, 3
- [11] A. Q. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, “GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models,” in *ICML*, 2022. 2
- [12] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *CVPR*, 2022. 2, 3, 4, 5, 6
- [13] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman et al., “LAION-5B: An open large-scale dataset for training next generation image-text models,” *NeurIPS*, 2022. 2, 3
- [14] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, D. Nair, S. Paul, W. Berman, Y. Xu, S. Liu, and T. Wolf, “Diffusers: State-of-the-art diffusion models,” *GitHub repository*, 2022. 2, 16
- [15] T. Brooks, A. Holynski, and A. A. Efros, “Instructpix2pix: Learning to follow image editing instructions,” in *CVPR*, 2023. 2, 5
- [16] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *ICCV*, 2023. 2
- [17] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, “DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation,” in *CVPR*, 2023. 2, 3
- [18] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt, and K. Schindler, “Repurposing diffusion-based image generators for monocular depth estimation,” in *CVPR*, 2024. 2, 3, 4, 7, 8, 9
- [19] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, “Depth anything v2,” *arXiv preprint arXiv:2406.09414*, 2024. 2, 3, 7
- [20] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *International Journal of Robotics Research*, 2013. 2
- [21] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, “Indoor segmentation and support inference from RGBD images,” in *ECCV*, 2012. 2, 6, 7, 10
- [22] S. Luo, Y. Tan, L. Huang, J. Li, and H. Zhao, “Latent consistency models: Synthesizing high-resolution images with few-step inference,” 2023. 2, 3, 12, 13
- [23] G. M. Garcia, K. A. Zeid, C. Schmidt, D. de Geus, A. Hermans, and B. Leibe, “Fine-tuning image-conditional diffusion models is easier than you think,” *arXiv preprint arXiv:2409.11355*, 2024. 2, 3, 4, 6, 7, 8, 9, 10, 13
- [24] M. Gui, J. S. Fischer, U. Prestel, P. Ma, D. Kotovenko, O. Grebenkova, S. A. Baumann, V. T. Hu, and B. Ommer, “Depthfm: Fast monocular depth estimation with flow matching,” *arXiv preprint arXiv:2403.13788*, 2024. 2, 3, 7
- [25] J. He, H. Li, W. Yin, Y. Liang, L. Li, K. Zhou, H. Liu, B. Liu, and Y.-C. Chen, “Lotus: Diffusion-based visual foundation model for high-quality dense prediction,” *arXiv preprint arXiv:2409.18124*, 2024. 2, 3, 4, 7, 9, 10
- [26] G. Xu, Y. Ge, M. Liu, C. Fan, K. Xie, Z. Zhao, H. Chen, and C. Shen, “Diffusion models trained with large data are transferable visual models,” *arXiv preprint arXiv:2403.06090*, 2024. 2, 3, 4, 7, 9, 10
- [27] O. B. Bohan, “Tiny autoencoder for stable diffusion,” <https://hf.co/madebyollin/taesd>, 2024, last accessed 15.06.2024. 2, 6, 7
- [28] B. K. Horn, “Shape from shading: A method for obtaining the shape of a smooth opaque object from one view,” 1970. 2, 4
- [29] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *NeurIPS*, 2020. 3, 5, 6
- [30] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in *ICLR*, 2021. 3, 6, 9, 10
- [31] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, “Consistency models,” in *ICML*, 2023. 3, 13
- [32] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in *ICLR*, 2021. 3
- [33] T. Wang, M. Kanakis, K. Schindler, L. Van Gool, and A. Obukhov, “Breathing new life into 3d assets with generative repainting,” in *BMVC*, 2023. 3
- [34] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson et al., “On the opportunities and risks of foundation models,” *arXiv preprint arXiv:2108.07258*, 2022. 3
- [35] H. Yao, R. Zhang, and C. Xu, “Visual-language prompt tuning with knowledge-guided context optimization,” in *CVPR*, 2023. 3
- [36] R. Zhang, X. Hu, B. Li, S. Huang, H. Deng, Y. Qiao, P. Gao, and H. Li, “Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners,” in *CVPR*, 2023. 3
- [37] H. Bahng, A. Jahanian, S. Sankaranarayanan, and P. Isola, “Exploring visual prompts for adapting large-scale models,” *arXiv preprint arXiv:2203.17274*, 2022. 3
- [38] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, and Y. Qiao, “CLIP-adapter: Better vision-language models with feature adapters,” *ICJY*, 2023. 3
- [39] R. Zhang, R. Fang, W. Zhang, P. Gao, K. Li, J. Dai, Y. Qiao, and H. Li, “Tip-adapter: Training-free CLIP-adapter for better vision-language modeling,” *arXiv:2111.03930*, 2021. 3
- [40] O. Pantazis, G. Brostow, K. Jones, and O. Mac Aodha, “SVL-adapter: Self-supervised adapter for vision-language pretrained models,” in *BMVC*, 2022. 3
- [41] C. Zhou, C. C. Loy, and B. Dai, “Extract free dense labels from CLIP,” in *ECCV*, 2022. 3
- [42] W. Zhao, Y. Rao, Z. Liu, B. Liu, J. Zhou, and J. Lu, “Unleashing text-to-image diffusion models for visual perception,” in *ICCV*, 2023. 3
- [43] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep ordinal regression network for monocular depth estimation,” in *CVPR*, 2018. 3
- [44] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, “From big to small: Multi-scale local planar guidance for monocular depth estimation,” *arXiv preprint arXiv:1907.10326*, 2019. 3
- [45] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, “NeWCRFs: Neural window fully-connected CRFs for monocular depth estimation,” in *CVPR*, 2022. 3
- [46] V. Patil, C. Sakaridis, A. Liniger, and L. Van Gool, “P3depth: Monocular depth estimation with a piecewise planarity prior,” in *CVPR*, 2022. 3
- [47] C. Liu, S. Kumar, S. Gu, R. Timofte, and L. Van Gool, “VA-DepthNet: A variational approach to single image depth prediction,” in *ICLR*, 2023. 3
- [48] S. F. Bhat, I. Alhashim, and P. Wonka, “AdaBins: Depth estimation using adaptive bins,” in *CVPR*, 2021. 3

- [49] Z. Li, X. Wang, X. Liu, and J. Jiang, "Binsformer: Revisiting adaptive bins for monocular depth estimation," *IEEE TIP*, vol. 33, pp. 3964–3976, 2024. **3**
- [50] J. Ning, C. Li, Z. Zhang, C. Wang, Z. Geng, Q. Dai, K. He, and H. Hu, "All in tokens: Unifying output space of visual tasks via soft token," in *ICCV*, 2023. **3**
- [51] G. Yang, H. Tang, M. Ding, N. Sebe, and E. Ricci, "Transformer-based attention networks for continuous pixel-wise prediction," in *ICCV*, 2021. **3**
- [52] Z. Li, Z. Chen, X. Liu, and J. Jiang, "Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation," *Machine Intelligence Research*, pp. 1–18, 2023. **3**
- [53] S. Aich, J. M. U. Vianney, M. A. Islam, M. Kaur, and B. Liu, "Bidirectional attention network for monocular depth estimation," in *ICRA*, 2021. **3**
- [54] S. F. Bhat, R. Birkel, D. Wofk, P. Wonka, and M. Müller, "ZoeDepth: Zero-shot transfer by combining relative and metric depth," *arXiv preprint arXiv:2302.12288*, 2023. **3, 14, 15**
- [55] W. Yin, C. Zhang, H. Chen, Z. Cai, G. Yu, K. Wang, X. Chen, and C. Shen, "Metric3D: Towards zero-shot metric 3d prediction from a single image," in *ICCV*, 2023. **3, 7**
- [56] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen, "Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation," *arXiv preprint arXiv:2404.15506*, 2024. **3, 7, 9**
- [57] V. Guizilini, I. Vasiljevic, D. Chen, R. Ambrus, and A. Gaidon, "Towards zero-shot scale-aware monocular depth estimation," in *ICCV*, 2023. **3**
- [58] Z. Li and N. Snavely, "MegaDepth: Learning single-view depth prediction from internet photos," in *CVPR*, 2018. **3**
- [59] W. Yin, X. Wang, C. Shen, Y. Liu, Z. Tian, S. Xu, C. Sun, and D. Renyin, "Diversedepth: Affine-invariant depth prediction using diverse data," *arXiv preprint arXiv:2002.00569*, 2020. **3, 7**
- [60] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE PAMI*, 2020. **3, 5, 7, 14, 15**
- [61] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *ICCV*, 2021. **3, 7**
- [62] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," in *CVPR*, 2024. **3, 7**
- [63] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby et al., "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024. **3, 4**
- [64] Y. Duan, X. Guo, and Z. Zhu, "DiffusionDepth: Diffusion denoising approach for monocular depth estimation," in *ECCV*, 2024. **3**
- [65] S. Saxena, A. Kar, M. Norouzi, and D. J. Fleet, "Monocular depth estimation using diffusion models," *arXiv preprint arXiv:2302.14816*, 2023. **3**
- [66] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M. A. Bautista, N. Paczan, R. Webb, and J. M. Susskind, "Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding," in *ICCV*, 2021. **3, 4, 6, 8, 10, 11**
- [67] Y. Cabon, N. Murray, and M. Humenberger, "Virtual KITTI 2," *arXiv preprint arXiv:2001.10773*, 2020. **3, 6, 8**
- [68] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *ICLR*, 2023. **3**
- [69] X. Fu, W. Yin, M. Hu, K. Wang, Y. Ma, P. Tan, S. Shen, D. Lin, and X. Long, "Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image," in *ECCV*, 2024. **3, 7, 9, 10**
- [70] X. Zhang, B. Ke, H. Riemenschneider, N. Metzger, A. Obukhov, M. Gross, K. Schindler, and C. Schroers, "Betterdepth: Plug-and-play diffusion refiner for zero-shot monocular depth estimation," *NeurIPS*, 2024. **3, 14**
- [71] J. Gregorek and L. Nalpantidis, "Steeredmarigold: Steering diffusion towards depth completion of largely incomplete depth maps," *arXiv preprint arXiv:2409.10202*, 2024. **3**
- [72] F. Tosi, P. Z. Ramirez, and M. Poggi, "Diffusion models for monocular depth estimation: Overcoming challenging conditions," in *ECCV*, 2024. **3**
- [73] Y. Jia, L. Hoyer, S. Huang, T. Wang, L. V. Gool, K. Schindler, and A. Obukhov, "Dginstyle: Domain-generalizable semantic segmentation with image diffusion models and stylized semantic control," in *ECCV*, 2024. **3, 4**
- [74] J. Shao, Y. Yang, H. Zhou, Y. Zhang, Y. Shen, M. Poggi, and Y. Liao, "Learning temporally consistent video depth from video diffusion priors," *arXiv preprint arXiv:2406.01493*, 2024. **3**
- [75] W. Hu, X. Gao, X. Li, S. Zhao, X. Cun, Y. Zhang, L. Quan, and Y. Shan, "Depthcrafter: Generating consistent long depth sequences for open-world videos," *arXiv preprint arXiv:2409.02095*, 2024. **3**
- [76] L. Ladický, B. Zeisl, and M. Pollefeys, "Discriminatively trained dense surface normal estimation," in *ECCV*, 2014. **3**
- [77] X. Wang, D. Fouhey, and A. Gupta, "Designing deep networks for surface normal estimation," in *CVPR*, 2015. **3**
- [78] J. Huang, Y. Zhou, T. Funkhouser, and L. Guibas, "Framenet: Learning local canonical frames of 3d surfaces from a single rgb image," in *ICCV*, 2019. **3, 10**
- [79] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *ICCV*, 2015. **3**
- [80] G. Bae, I. Budvytis, and R. Cipolla, "Estimating and exploiting the aleatoric uncertainty in surface normal estimation," in *ICCV*, 2021. **3**
- [81] O. F. Kar, T. Yeo, A. Atanov, and A. Zamir, "3d common corruptions and data augmentation," in *CVPR*, 2022. **3, 9**
- [82] G. Bae and A. J. Davison, "Rethinking inductive biases for surface normal estimation," in *CVPR*, 2024. **3, 9, 10**
- [83] C. Ye, L. Qiu, X. Gu, Q. Zuo, Y. Wu, Z. Dong, L. Bo, Y. Xiu, and X. Han, "Stablenormal: Reducing diffusion variance for stable and sharp normal," *ACM Transactions on Graphics (TOG)*, 2024. **4, 9, 10**
- [84] H. Barrow, J. Tenenbaum, A. Hanson, and E. Riseman, "Recovering intrinsic scene characteristics," *Comput. vis. syst.*, vol. 2, no. 3-26, p. 2, 1978. **4**
- [85] Z. Li, M. Shafiei, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker, "Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image," in *CVPR*, 2020. **4**
- [86] Z. Wang, J. Philion, S. Fidler, and J. Kautz, "Learning indoor inverse rendering with 3d spatially-varying lighting," in *ICCV*, 2021. **4**
- [87] Z. Li, J. Shi, S. Bi, R. Zhu, K. Sunkavalli, M. Hašan, Z. Xu, R. Ramamoorthi, and M. Chandraker, "Physically-based editing of indoor scene lighting from a single image," in *ECCV*, 2022. **4**
- [88] J. Luo, D. Ceylan, J. S. Yoon, N. Zhao, J. Philip, A. Frühstück, W. Li, C. Richardt, and T. Wang, "Intrinsicdiffusion: Joint intrinsic layers from latent diffusion models," in *ACM SIGGRAPH*, 2024, pp. 1–11. **4**
- [89] R. Zhu, Z. Li, J. Matai, F. Porikli, and M. Chandraker, "Irisformer: Dense vision transformers for single-image inverse rendering in indoor scenes," in *CVPR*, 2022. **4**
- [90] C. Careaga and Y. Aksoy, "Colorful diffuse intrinsic image decomposition in the wild," *ACM Trans. Graph.*, vol. 43, no. 6, 2024. **4, 11, 12**
- [91] —, "Intrinsic image decomposition via ordinal shading," *ACM Transactions on Graphics*, vol. 43, no. 1, pp. 1–24, 2023. **4**
- [92] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019. **4**
- [93] A. Bhattad, D. McKee, D. Hoiem, and D. Forsyth, "Stylegan knows normal, depth, albedo, and more," *NeurIPS*, 2024. **4**
- [94] X. Du, N. Kolkin, G. Shakhnarovich, and A. Bhattad, "Generative models: What do they know? do they know things? let's find out!" *arXiv preprint arXiv:2311.17137*, 2023. **4**
- [95] H.-Y. Lee, H.-Y. Tseng, and M.-H. Yang, "Exploiting diffusion prior for generalizable dense prediction," in *CVPR*, 2024. **4**
- [96] P. Kocsis, V. Sitzmann, and M. Nießner, "Intrinsic image diffusion for indoor single-view material estimation," in *CVPR*, 2024. **4, 11, 12, 16**
- [97] Z. Zeng, V. Deschaintre, I. Georgiev, Y. Hold-Geoffroy, Y. Hu, F. Luan, L.-Q. Yan, and M. Hašan, "Rgb-x: Image decomposition and synthesis using material-and lighting-aware diffusion models," in *SIGGRAPH conference*, 2024. **4, 11, 12**
- [98] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., "Learning transferable visual models from natural language supervision," in *ICML*, 2021, pp. 8748–8763. **4**
- [99] J. Zhu, F. Luan, Y. Huo, Z. Lin, Z. Zhong, D. Xi, R. Wang, H. Bao, J. Zheng, and R. Tang, "Learning-based inverse rendering of complex indoor scenes with differentiable monte carlo raytracing," in *SIGGRAPH Asia conference. ACM*, 2022. **4, 10, 11**
- [100] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris, and Y. Aksoy, "Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging," in *CVPR*, 2021. **4, 14, 15**
- [101] Z. Li, S. F. Bhat, and P. Wonka, "Patchfusion: An end-to-end tile-based framework for high-resolution monocular metric depth estimation," in *CVPR*, 2024. **4, 14, 15**
- [102] —, "Patchrefiner: Leveraging synthetic data for real-domain high-resolution monocular metric depth estimation," in *ECCV*, 2024. **4**
- [103] O. Bar-Tal, L. Yariv, Y. Lipman, and T. Dekel, "Multidiffusion: fusing diffusion paths for controlled image generation," in *ICML*, 2023. **4, 14**



- [104] A. Bochkovskii, A. Delaunoy, H. Germain, M. Santos, Y. Zhou, S. R. Richter, and V. Koltun, "Depth pro: Sharp monocular metric depth in less than a second," *arXiv*, 2024. [4](#), [14](#), [15](#)
- [105] W. Wagner, A. Ullrich, V. Ducic, T. Melzer, and N. Studnicka, "Gaussian decomposition and calibration of a novel small-footprint full-waveform digitising airborne laser scanner," *ISPRS journal of Photogrammetry and Remote Sensing*, vol. 60, no. 2, pp. 100–112, 2006. [5](#)
- [106] S. Huang, Z. Gojcic, Z. Wang, F. Williams, Y. Kasten, S. Fidler, K. Schindler, and O. Litany, "Neural lidar fields for novel view synthesis," in *ICCV*, 2023. [5](#)
- [107] A. Eftekhar, A. Sax, J. Malik, and A. Zamir, "Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans," in *ICCV*, 2021. [5](#), [7](#)
- [108] T. Salimans and J. Ho, "Progressive distillation for fast sampling of diffusion models," in *ICLR*, 2022. [6](#)
- [109] W. Yin, J. Zhang, O. Wang, S. Niklaus, L. Mai, S. Chen, and C. Shen, "Learning to recover 3d scene shape from a single image," in *CVPR*, 2021. [7](#), [14](#), [15](#)
- [110] C. Zhang, W. Yin, B. Wang, G. Yu, B. Fu, and C. Shen, "Hierarchical normalization for robust monocular depth estimation," *NeurIPS*, 2022. [7](#)
- [111] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *CVPR*, 2012. [6](#)
- [112] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3d reconstructions of indoor scenes," in *CVPR*, 2017. [6](#), [10](#)
- [113] T. Schops, J. L. Schonberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *CVPR*, 2017. [7](#)
- [114] I. Vasiljevic, N. Kolkin, S. Zhang, R. Luo, H. Wang, F. Z. Dai, A. F. Daniele, M. Mostajabi, S. Basart, M. R. Walter, and G. Shakhnarovich, "DIODE: A Dense Indoor and Outdoor DDepth Dataset," *arXiv preprint arXiv:1908.00463*, 2019. [7](#), [10](#)
- [115] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *ECCV*, 2012. [10](#)
- [116] T. Koch, L. Liebel, F. Fraundorfer, and M. Körner, "Evaluation of cnn-based single-image depth estimation methods," in *ECCV workshop*, 2019. [10](#)
- [117] W. Chen, S. Qian, D. Fan, N. Kojima, M. Hamilton, and J. Deng, "OASIS: A large-scale dataset for single image 3d in the wild," in *CVPR*, 2020. [10](#)
- [118] A. Bansal, B. Russell, and A. Gupta, "Marr Revisited: 2D-3D model alignment via surface normal prediction," in *CVPR*, 2016. [10](#)
- [119] D. F. Fouhey, A. Gupta, and M. Hebert, "Data-driven 3D primitives for single image understanding," in *ICCV*, 2013. [10](#)
- [120] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE TIP*, vol. 13, no. 4, pp. 600–612, 2004. [11](#)
- [121] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018. [11](#)
- [122] Y. Song and P. Dhariwal, "Improved techniques for training consistency models," in *ICLR*, 2024. [13](#)
- [123] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019. [13](#)
- [124] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *GCPR*. Springer, 2014. [14](#)
- [125] P. Z. Ramirez, F. Tosi, L. Di Stefano, R. Timofte, A. Costanzino, M. Poggi, S. Salti, S. Mattoccia, Y. Zhang, C. Wu et al., "Ntire 2024 challenge on hr depth from images of specular and transparent surfaces," in *CVPR*, 2024. [14](#)
- [126] T. Koch, L. Liebel, F. Fraundorfer, and M. Korner, "Evaluation of cnn-based single-image depth estimation methods," in *ECCV workshop*, 2018. [14](#)
- [127] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries," in *WACV*, 2019. [14](#)

**Bingxin Ke** is a PhD student at the Photogrammetry and Remote Sensing Lab of ETH Zurich. His interest lies in generalizable computer vision, especially for 3D vision problems. He earned his Bachelor's degree in Geomatics Engineering at Wuhan University in 2020, and Master's degree in Geomatics at ETH Zurich in 2022.

**Kevin Qu** is a Master student at the Photogrammetry and Remote Sensing Lab of ETH Zurich. He obtained his Bachelor's degree in Electrical and Computer Engineering at the Technical University of Munich in 2023, and is currently pursuing his Master's degree in Robotics, Systems and Control at ETH. His research interests include generative models, 3D vision and robotic perception.

**Tianfu Wang** is a PhD student at the Intelligent Sensing Lab of the University of Maryland, College Park. Tianfu completed his Master's degree in Computer Science at ETH Zurich, where he worked in the Photogrammetry and Remote Sensing Lab. Prior to that, Tianfu earned his Bachelor's degree in Computer Science and Mathematics from Northwestern University. Tianfu is interested in computational imaging, generative models, and differentiable rendering.

**Nando Metzger** is a PhD student at the Photogrammetry and Remote Sensing Lab of ETH Zurich. He is interested in weakly supervised learning and super-resolution techniques and their applications to monocular depth and remote sensing. He studied Geomatics at ETH Zurich, where he received his Bachelor's degree in 2019 and his Master's degree in 2021.

**Shengyu Huang** is a PhD student at the Photogrammetry and Remote Sensing Lab of ETH Zurich. He is interested in 3D vision problems and applications of neural fields beyond conventional cameras. He earned a Bachelor's degree in Surveying and Mapping Engineering at Tongji University in 2018, later he obtained his Master's degree in Geomatics from ETH Zurich in 2020.

**Bo Li** is a student collaborator with the Photogrammetry and Remote Sensing Lab of ETH Zurich. His research interests include computer graphics, generative models and efficient GPU computation for large-scale AI workloads. He earned dual Bachelor's degrees in Biological Science and Computer Science from Peking University and Master's degree in Computer Science at ETH Zurich.

**Anton Obukhov** is an established researcher in the Photogrammetry and Remote Sensing Lab, broadly interested in computer vision and machine learning research. He joined the Computer Vision Laboratory at ETH Zurich in 2018 and received his PhD in 2022, supervised by Prof. Luc Van Gool and funded by the Toyota TRACE-Zurich project. He also holds a Diploma in Computational Mathematics and Cybernetics from Moscow State University, obtained in 2008. Between these degrees, he spent a decade working in the industry: he helped NVIDIA drive the adoption of NVIDIA CUDA technology in scientific computing and later helped Ubiquiti Networks build multiple video camera products with varying degree of artificial intelligence.

**Konrad Schindler** (Senior Member, IEEE) received the Diplomingenieur (MTech) degree from the Vienna University of Technology, Vienna, Austria, in 1999, and the PhD from the Graz University of Technology, Graz, Austria, in 2003. He was a photogrammetric engineer in the private industry and held researcher positions with the Graz University of Technology, Monash University, Melbourne, VIC, Australia, and ETH Zürich, Zürich, Switzerland. He was an assistant professor of image understanding with TU Darmstadt, Darmstadt, Germany, in 2009. Since 2010, he has been a tenured professor of photogrammetry and remote sensing with ETH Zürich. His research interests include computer vision, photogrammetry, and remote sensing.