BetterDepth: Plug-and-Play Diffusion Refiner for Zero-Shot Monocular Depth Estimation

Xiang Zhang[1,2], **Bingxin Ke**[1], **Hayko Riemenschneider**[2], **Nando Metzger**[1],
Anton Obukhov[1], **Markus Gross**[1,2], **Konrad Schindler**[1], **Christopher Schroers**[2]
[1]ETH Zürich,   [2]DisneyResearch|Studios

Abstract

By training over large-scale datasets, zero-shot monocular depth estimation (MDE) methods show robust performance in the wild but often suffer from insufficient detail. Although recent diffusion-based MDE approaches exhibit a superior ability to extract details, they struggle in geometrically complex scenes that challenge their geometry prior, trained on less diverse 3D data. To leverage the complementary merits of both worlds, we propose *BetterDepth* to achieve geometrically correct affine-invariant MDE while capturing fine details. Specifically, BetterDepth is a conditional diffusion-based refiner that takes the prediction from pre-trained MDE models as depth conditioning, in which the global depth layout is well-captured, and iteratively refines details based on the input image. For the training of such a refiner, we propose global pre-alignment and local patch masking methods to ensure BetterDepth remains faithful to the depth conditioning while learning to add fine-grained scene details. With efficient training on small-scale synthetic datasets, BetterDepth achieves state-of-the-art zero-shot MDE performance on diverse public datasets and on in-the-wild scenes. Moreover, BetterDepth can improve the performance of other MDE models in a plug-and-play manner without further re-training.

1 Introduction

As a fundamental task in computer vision, monocular depth estimation (MDE) aims to extract depth information from single-view images, benefitting various real-world applications [46, 54, 48, 27]. Unlike traditional depth estimation techniques that utilize geometric relationships from stereo [21] or structured light setups [42], MDE is a highly ill-posed task and relies on the geometric prior knowledge learned from training datasets, where real data plays a pivotal role to ensure generalization to in-the-wild applications [31, 30, 49]. However, due to the difficulty of collecting fine-grained depth labels in real scenarios, real-world depth labels are often coarse, noisy, and incomplete, resulting in a trade-off between the quality and generalization of MDE. Thus, although significant progress in zero-shot MDE has been achieved with techniques like mixing diverse training datasets [31] and unleashing large-scale unlabeled data [49], previous MDE approaches often suffer from over-smoothing of details, as indicated by the red arrows in Fig. 1.

Recently, diffusion models have exhibited promising performance in a variety of computer vision tasks [13, 44, 22, 47, 32], including MDE [39, 17, 12, 10]. Benefitting from the iterative refinement scheme, diffusion-based MDE methods can produce impressive depth maps with fine granularity as depicted in Fig. 1. However, training a diffusion-based MDE generally requires complete depth labels [17, 10, 39], which is in practice achieved by rendering synthetic datasets. Compared to real data, existing synthetic RGB-D datasets exhibit lower variety and contain fewer samples which limits the generality of the learned prior. Despite several attempts to improve the generalization of diffusion-based MDE, such as label infilling [39] and transferring 2D image priors [17], current
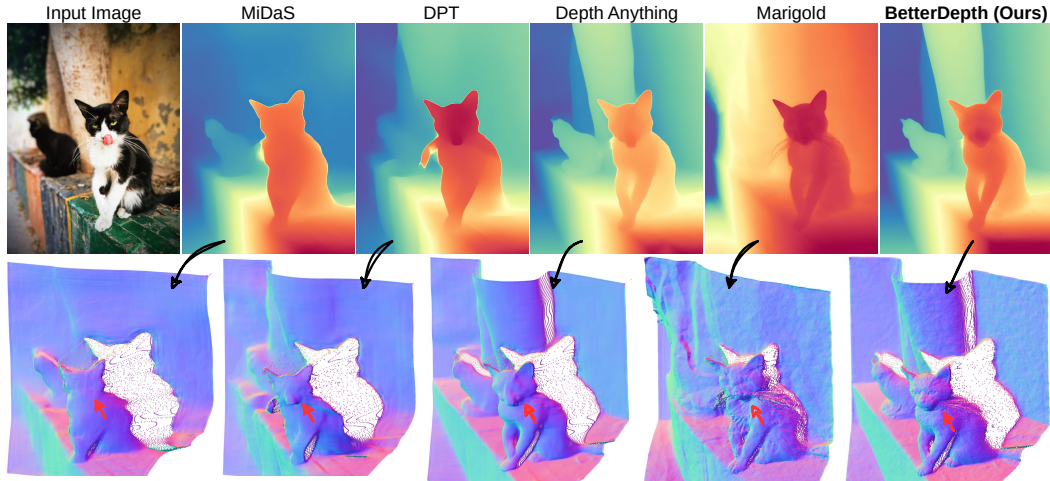
Figure 1: **Monocular depth estimation** (depth map and 3D reconstruction with color-coded normals). Feed-forward methods, like Depth Anything [49], produce robust global 3D shape but suffer from over-smoothed details. Diffusion-based methods, like Marigold [17], extract fine details but fall short in zero-shot global shape recovery. Our proposed BetterDepth offers the best of both worlds and achieves robust zero-shot depth estimation with fine details.

diffusion-based approaches still have relatively limited a-priori knowledge of global layout. This results in less accurate predictions in challenging scenes compared to models trained with diverse datasets, *e.g.*, Depth Anything [49] (Tab. 2).

In this work, we aim for robust affine-invariant MDE while also capturing fine-grained details. Motivated by the complementary merits of feed-forward and diffusion-based MDE methods, we propose *BetterDepth* to boost pre-trained MDE models with a diffusion refiner, simultaneously leveraging rich geometric priors for zero-shot transfer and diffusion models for detail refinement. Specifically, BetterDepth is designed as a depth-conditioned diffusion model to retain the zero-shot generalization power of pre-trained MDE models. Through efficient training on small-scale synthetic datasets, BetterDepth further attains a remarkable ability to extract details (Fig. 1) and can directly improve other MDE models, without re-training. To learn detail refinement and simultaneously preserve the prior knowledge from pre-trained MDE models, we introduce global pre-alignment and local patch masking strategies during training, to ensure the faithfulness of BetterDepth to depth conditioning while enabling fine-grained detail extraction. In this way, BetterDepth combines the advantages of zero-shot and diffusion-based MDE models, exhibiting state-of-the-art performance and producing visually superior results on diverse datasets. Our main contributions are:

- We propose BetterDepth to boost zero-shot MDE methods with a plug-and-play diffusion refiner, achieving robust affine-invariant MDE performance with fine-grained details.
- We design global pre-alignment and local patch masking strategies to enable learning the refinement from small-scale synthetic datasets while preserving the rich prior knowledge in pre-trained zero-shot MDE models.

2   Related Work

Zero-Shot Monocular Depth Estimation. A variety of attempts are devoted to improving the robustness of MDE in the wild, *i.e.*, zero-shot depth estimation, which aims to predict depth for any input image taken in unconstrained settings [3, 4, 53, 55, 52, 15, 29]. Considering that MDE is a geometrically ill-posed problem, many zero-shot MDE works are designed to estimate affine-invariant depth, *i.e.*, predicting the depth values up to an unknown global scale and shift [31, 17, 53, 12, 49]. For example, MegaDepth [23] and DiverseDepth [51] collect internet images for network training, improving adaptation to unseen scenes. Furthermore, MiDaS [31] proposes a family of scale- and shift-invariant losses to handle the different depth representations, *e.g.*, metric depth and inverse depth (disparity), across datasets, so as to mix diverse training datasets and reach robust zero-shot

transfer. By replacing CNN backbones with powerful vision transformers, DPT [30] and Omnidata [8] further boost the performance of zero-shot depth estimation. Recently, Depth Anything developed a semi-supervised strategy to unleash the power of large-scale unlabeled images (62M) and acquire a robust representation for in-the-wild prediction [49]. Although the zero-shot generalization of MDE grows with the amount of training data, the lower-quality labels in real-world datasets tend to hinder the reconstruction of fine-grained depth details, resulting in over-smoothing as shown in Fig. 1.

Diffusion-Based Monocular Depth Estimation. The emergence of denoising diffusion probabilistic models (DDPMs) brought up a new paradigm for image generation, producing high-quality images with realistic details [13, 44, 34]. Many works have showcased the effectiveness of diffusion models in generating photo-realistic results for various computer vision tasks [37, 22, 35, 25, 5]. In the realm of MDE, DDP [16] describes a diffusion-based framework for dense visual prediction tasks, and DiffusionDepth [7] further utilizes Swin transformers [24] for image encoding, performing iterative refinement in the depth latent space. Considering the noisy and sparse depth labels in practice, several techniques are proposed, *e.g.*, depth infilling [40] and self-supervised pre-training [39], to achieve better MDE performance. A recently emerging trend is to exploit the prior knowledge in foundational diffusion models for MDE [56, 17, 12]. Marigold [17] proposes an efficient fine-tuning protocol to leverage the rich prior in the Stable Diffusion model [34] for depth estimation, producing visually compelling depth results. Following this direction, DepthFM [12] improves inference speed with flow matching, and GeoWizard [10] utilizes cross-modal relations for joint depth and normal prediction. However, existing diffusion-based approaches still struggle to outperform the feed-forward MDE models like Depth Anything [49] (Tab. 2), due to the difficulty of learning diverse geometric depth priors from datasets with few or sparse depth labels [39]. By contrast, our BetterDepth efficiently leverages the rich prior knowledge of feed-forward models and improves the extraction of details with diffusion, achieving state-of-the-art MDE performance (Tab. 2) with compelling visual results (Figs. 1 and 5).

3   Method

We first analyze existing MDE methods and formulate our objective in Sec. 3.1. Based on the analysis, we then propose our BetterDepth framework in Sec. 3.2 and introduce the training and inference strategies designed specifically for BetterDepth in Sec. 3.3 and 3.4, respectively.

3.1   Problem Formulation

Model architecture and training data are two key factors that determine MDE performance. Given a depth dataset $\{(\mathbf{x}_i, \mathbf{d}_i)\}_i \in \mathbf{D}$ with $\mathbf{x}_i$ and $\mathbf{d}_i$ corresponding to images and depth labels, previous zero-shot MDE approaches usually employ feed-forward models $\mathbf{M}_{\mathrm{FFD}}$ and learn depth estimation using the following training objective [31, 30, 49]:

$$\mathcal{L}_{\mathrm{MDE}}\big(\mathbf{d}_i, \mathbf{M}_{\mathrm{FFD}}(\mathbf{x}_i)\big), \tag{1}$$

where $\mathcal{L}_{\mathrm{MDE}}(\cdot)$ represents a suitable loss function, *e.g.*, scale- and shift-invariant losses [31]. Since $\mathbf{d}_i$ is only used to supervise model outputs in Eq. (1), feed-forward MDE methods can easily handle invalid pixels in depth labels via techniques like masking, and thus gain robust zero-shot capability by learning from diverse large-scale datasets [31, 30, 49]. To handle the synthetic-to-real domain gaps caused by synthetic data $\mathbf{D}_{\mathrm{syn}}$ [1], real-world datasets $\mathbf{D}_{\mathrm{real}}$ are often simultaneously employed to learn more robust representations for in-the-wild prediction. However, the quality of depth labels in $\mathbf{D}_{\mathrm{real}}$ usually hinders feed-forward methods from learning to capture high-frequency information present in the inputs, resulting in over-smoothed details, as depicted in Fig. 1.

By contrast, diffusion-based MDE approaches generally excel at capturing fine-grained details via iterative refinement [17, 10]. Different from feed-forward methods, diffusion models $\mathbf{M}_{\mathrm{DM}}$ comprise a T-step forward process to gradually corrupt samples with Gaussian noise at each timestamp $t \in \{1, \ldots, T\}$, and a learned reverse process to transform random Gaussian noise to a sample from the target data distribution [13, 44]. Instead of directly fitting $\mathbf{d}_i$ in Eq. (1), one typically learns to estimate the added Gaussian noise from $\mathbf{x}_i$ and $\mathbf{d}_i$ at each timestamp t, *i.e.*:

$$\mathcal{L}_{\mathrm{DM}}\big(\boldsymbol{\epsilon}, \mathbf{M}_{\mathrm{DM}}\left(\mathbf{x}_i, \mathrm{AddNoise}(\mathbf{d}_i, \boldsymbol{\epsilon}, t)\right)\big), \tag{2}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ denotes Gaussian noise; $\mathrm{AddNoise}(\cdot)$ is an operator that corrupts depth labels $\mathbf{d}_i$ with noise $\boldsymbol{\epsilon}$ according to t; $\mathcal{L}_{\mathrm{DM}}(\cdot)$ represents a loss function for diffusion models, like the velocity

Table 1: **Performance comparison** between feed-forward and diffusion-based MDE. $\mathbf{M}_{\mathrm{FFD}}$ and $\mathbf{M}_{\mathrm{DM}}$ correspond to feed-forward and diffusion-based architectures, respectively. $\mathbf{D}_{\mathrm{syn}}$ and $\mathbf{D}_{\mathrm{real}}$ denote synthetic and real datasets, respectively. $\mathcal{X}(\mathbf{M}, \mathbf{D})$ is the output distribution with a selected model $\mathbf{M}$ and training set $\mathbf{D}$. Our goal is to approach the ideal distribution $\mathcal{X}(\mathbf{M}_{\mathrm{ideal}}, \mathbf{D}_{\mathrm{ideal}})$ and achieve zero-shot MDE with precise details.

Model	Training Data	Output Distribution	Fine-Grained Details	Zero-Shot Generalizability
$\mathbf{M}_{\mathrm{FFD}}$	$\mathbf{D}_{\mathrm{syn}}, \mathbf{D}_{\mathrm{real}}$	$\mathcal{X}(\mathbf{M}_{\mathrm{FFD}}, \{\mathbf{D}_{\mathrm{syn}}, \mathbf{D}_{\mathrm{real}}\})$		✓
$\mathbf{M}_{\mathrm{DM}}$	$\mathbf{D}_{\mathrm{syn}}$†	$\mathcal{X}(\mathbf{M}_{\mathrm{DM}}, \mathbf{D}_{\mathrm{syn}})$	✓	
$\mathbf{M}_{\mathrm{ideal}}$	$\mathbf{D}_{\mathrm{ideal}}$	$\mathcal{X}(\mathbf{M}_{\mathrm{ideal}}, \mathbf{D}_{\mathrm{ideal}})$	✓	✓

†We focus on diffusion-based MDE methods that are trained on synthetic data, due to their superior reconstruction of fine details.

prediction loss [38]. Since the depth labels are treated as model inputs in Eq. (2), directly training $\mathbf{M}_{\mathrm{DM}}$ with sparse depth labels becomes challenging [39], preventing training with diverse real-world data, thus limiting the generalization ability of diffusion-based MDE.

Based on the above analysis, we summarize the characteristics of feed-forward and diffusion-based MDE methods in Tab. 1, where $\mathcal{X}(\mathbf{M}, \mathbf{D})$ represents the output distribution, as a function of the employed model architecture $\mathbf{M}$ and training datasets $\mathbf{D}$. Motivated by the complementary strengths of $\mathcal{X}(\mathbf{M}_{\mathrm{FFD}}, \{\mathbf{D}_{\mathrm{syn}}, \mathbf{D}_{\mathrm{real}}\})$ and $\mathcal{X}(\mathbf{M}_{\mathrm{DM}}, \mathbf{D}_{\mathrm{syn}})$, our goal is to approach the ideal distribution $\mathcal{X}(\mathbf{M}_{\mathrm{ideal}}, \mathbf{D}_{\mathrm{ideal}})$ and achieve robust zero-shot MDE with fine-grained details. However, to reach this in a tractable manner, challenges exist from both the model and data perspectives:

- **Model Limitation.** A potential solution is to train diffusion models over diverse datasets, *i.e.*, $\mathbf{M}_{\mathrm{ideal}} = \mathbf{M}_{\mathrm{DM}}$ and $\mathbf{D}_{\mathrm{ideal}} = \{\mathbf{D}_{\mathrm{syn}}, \mathbf{D}_{\mathrm{real}}\}$. However, how to efficiently train $\mathbf{M}_{\mathrm{DM}}$ with $\mathbf{D}_{\mathrm{real}}$ while preserving the functionality to extract fine-grained details remains an open question. In addition, training over large datasets is required to gain robust zero-shot generalization, which would be extremely time-consuming and resource-intensive.

- **Data Limitation.** Another possible method is to train feed-forward models $\mathbf{M}_{\mathrm{FFD}}$ with high-quality diverse datasets. However, although high-quality labels are available in $\mathbf{D}_{\mathrm{syn}}$, training solely with $\mathbf{D}_{\mathrm{syn}}$ introduces a detrimental synthetic-to-real domain gap [1]. Meanwhile, real depth labels in $\mathbf{D}_{\mathrm{real}}$ must be collected with depth sensors like ToF cameras or LiDAR [11], which inherently limits the achievable quality of the supervision.

3.2 BetterDepth Framework

To circumvent the aforementioned limitations, we propose BetterDepth to efficiently leverage the strengths of feed-forward and diffusion-based methods, achieving better MDE performance. Specifically, BetterDepth is composed of a conditional latent diffusion model and a pre-trained feed-forward MDE model, as illustrated in Fig. 2. Since $\mathbf{M}_{\mathrm{FFD}}$ is known to reach strong zero-shot generalization by training on large and diverse datasets, we first utilize the rich geometric prior from pre-trained $\mathbf{M}_{\mathrm{FFD}}$, *e.g.*, DPT [30] or Depth Anything [49], to ensure accurate estimation of the global depth context. Based on this, a learnable $\mathbf{M}_{\mathrm{DM}}$ is employed to locally improve the estimation of details via iterative refinement. To enable the processing of high-resolution images, we follow Marigold [17] and implement $\mathbf{M}_{\mathrm{DM}}$ with Stable Diffusion [34], which maps from pixel space to a lower-dimensional latent space with a variational autoencoder (VAE) [19] and performs denoising with a U-Net in latent space. Because we treat $\mathbf{M}_{\mathrm{FFD}}$ as knowledge reservoir for zero-shot generalization and only need to train $\mathbf{M}_{\mathrm{DM}}$ for refinement, BetterDepth only requires a small synthetic training dataset, *e.g.*, 400 data pairs as shown in Tab. 2. Furthermore, the trained $\mathbf{M}_{\mathrm{DM}}$ in BetterDepth can be directly transferred to improve other $\mathbf{M}_{\mathrm{FFD}}$ models, without re-training.

3.3 Training Strategies

The training pipeline of BetterDepth is illustrated in Fig. 2. Although the pre-trained model $\mathbf{M}_{\mathrm{FFD}}$ in BetterDepth provides coarse depth estimates as reliable conditioning, directly training the diffusion-based refiner $\mathbf{M}_{\mathrm{DM}}$ with synthetic data still tends to overfit the training data distribution, resulting in similar performance as $\mathcal{X}(\mathbf{M}_{\mathrm{DM}}, \mathbf{D}_{\mathrm{syn}})$ and degarding generalization. To enhance the faithfulness of BetterDepth to the depth conditioning while still enabling refinement of details, we modify the
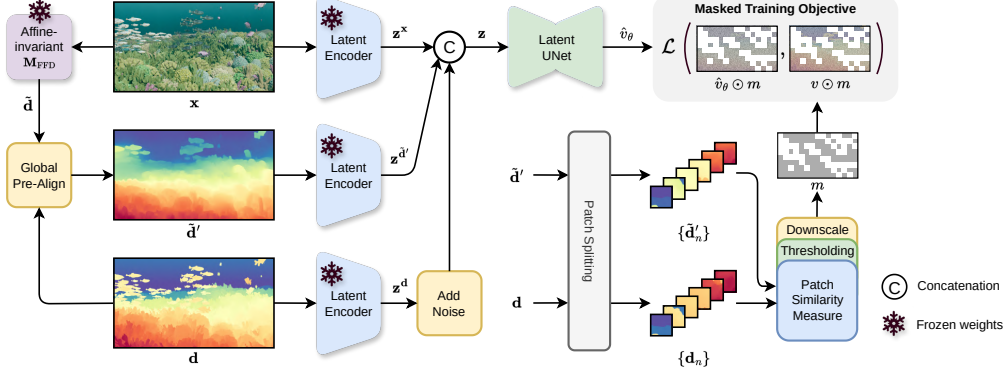
Figure 2: **BetterDepth training pipeline.** Given training images $\mathbf{x}$ and labels $\mathbf{d}$, we first estimate coarse depth maps $\tilde{\mathbf{d}}$ with the pre-trained $\mathbf{M}_{\text{FFD}}$ and apply global pre-alignment to $\tilde{\mathbf{d}}$ using $\mathbf{d}$ as reference. Afterwards, the frozen latent encoder is employed to convert the image $\mathbf{x}$, the depth labels $\mathbf{d}$, and the aligned depth conditioning $\tilde{\mathbf{d}}'$ to the latent space. To construct the masked training objective, $\tilde{\mathbf{d}}'$ and $\mathbf{d}$ are split into non-overlapping patches $\{\tilde{\mathbf{d}}'_n\}$ and $\{\mathbf{d}_n\}$, and dissimilar patches are filter out by thresholding, producing the patch-level similarity mask. Finally, the mask is downscaled to the latent space resolution for diffusion training.

diffusion training pipeline to include global pre-alignment and local patch masking techniques, simultaneously promoting zero-shot MDE capability and fine-grained detail extraction.

Global Pre-Alignment. To alleviate overfitting, we first propose a global pre-alignment method to narrow the gap between the conditioning depth map and the ground truth depth, enforcing BetterDepth to follow depth conditioning at a global scale. Given a pre-trained affine-invariant depth model $\mathbf{M}_{\text{FFD}}$ and a data pair $(\mathbf{x},\ \mathbf{d}) \in \mathbf{D}_{\text{syn}}$ (subscript i is omitted for brevity), we first estimate a coarse depth map $\tilde{\mathbf{d}}$ via $\tilde{\mathbf{d}} = \mathbf{M}_{\text{FFD}}(\mathbf{x})$ as depicted in Fig. 2. Although $\tilde{\mathbf{d}}$ and $\mathbf{d}$ correspond to the same image $\mathbf{x}$, the estimated depth values in $\tilde{\mathbf{d}}$ generally deviate from $\mathbf{d}$ due to the unknown scale and shift, which stops BetterDepth from establishing a strong dependency between the depth conditioning and the final estimate during training. We resolve this with a global pre-alignment to eliminate the difference caused by the unknown scale and shift. Inspired by the affine-invariant depth evaluation protocol [31], we first estimate the scale s and shift b and then align $\tilde{\mathbf{d}}$ to the depth labels $\mathbf{d}$, *i.e.*,

$$\tilde{\mathbf{d}}' = s\tilde{\mathbf{d}} + b, \text{ where } (s, b) = \arg\min_{s,b} \left\| s\tilde{\mathbf{d}} + b - \mathbf{d} \right\|_2^2. \tag{3}$$

Eq. (3) is solved via least squares fitting and $\tilde{\mathbf{d}}'$ indicates the aligned depth conditioning. Afterwards, the frozen latent VAE encoder is employed to project $\mathbf{x}$, $\tilde{\mathbf{d}}'$, $\mathbf{d}$ to latent space, corresponding to $\mathbf{z}^{\mathbf{x}}$, $\mathbf{z}^{\tilde{\mathbf{d}}'}$, $\mathbf{z}^{\mathbf{d}}$. We then follow the DDPM training scheme [13] to generate a noisy sample $\mathbf{z}_t^{\mathbf{d}} = \sqrt{\bar{\alpha}_t}\mathbf{z}_0^{\mathbf{d}} + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}$ with Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathbf{z}_0^{\mathbf{d}} := \mathbf{z}^{\mathbf{d}}$, $\bar{\alpha}_t := \prod_{j=1}^{t} 1 - \beta_j$, and $\{\beta_1, \ldots, \beta_T\}$ is the variance schedule of a T-step process. Finally, the noisy sample $\mathbf{z}_t^{\mathbf{d}}$ is concatenated with the latent image and depth conditioning $\mathbf{z}^{\mathbf{x}}$, $\mathbf{z}^{\tilde{\mathbf{d}}'}$ as inputs to train the latent U-Net.

Although our pre-alignment strengthens the conditioning by ensuring a similar global depth range between $\tilde{\mathbf{d}}'$ and the depth label $\mathbf{d}$, misalignment still exists in local regions due to the estimation bias of the pre-trained MDE model. Even though rectifying the coarse depth conditioning $\tilde{\mathbf{d}}'$ to the high-quality label $\mathbf{d}$ during training might intuitively seem helpful to MDE performance, we find that rectifying significantly different local regions between $\tilde{\mathbf{d}}'$ and $\mathbf{d}$ also degrades the zero-shot performance. This is because the pre-trained depth model embeds rich prior knowledge of the visual world, which is more important than the dataset-specific knowledge learned in small-scale training sets. Thus, we next propose local patch masking to further improve the efficacy of depth conditioning in local regions while learning detail refinement.

Local Patch Masking. As shown in Fig. 2, we first estimate the latent space mask m from depth label $\mathbf{d}$ and the aligned depth conditioning $\tilde{\mathbf{d}}'$, and then construct a masked diffusion objective for training. In detail, $\tilde{\mathbf{d}}'$ and $\mathbf{d}$ are first split into non-overlapping local patches $\{\tilde{\mathbf{d}}'_n\}$, $\{\mathbf{d}_n\}$, where $\tilde{\mathbf{d}}'_n \in \mathbb{R}^{w \times w}$

and $\mathbf{d}_n \in \mathbb{R}^{w \times w}$, with w the patch size. For each pair of patches we measure the similarity using the Euclidean distance, *i.e.*,

$$\mathrm{Dist}(\tilde{\mathbf{d}}'_n, \mathbf{d}_n) = \left\| \tilde{\mathbf{d}}'_n - \mathbf{d}_n \right\|_2, \tag{4}$$

and then generate the pixel space mask M by

$$M_n = \begin{cases} 1, & \text{if } \mathrm{Dist}(\tilde{\mathbf{d}}'_n, \mathbf{d}_n) \leq w \cdot \eta, \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where η indicates the average tolerance per pixel in the patch and controls the trade-off between depth conditioning and refinement of details. To fit the latent diffusion scheme, the pixel space mask M is then downscaled to a latent space mask m via $m = \mathrm{MaxPool}(M)$. Finally, m is applied to the velocity prediction objective [38] for model training,

$$\mathcal{L} = \mathbb{E}_{\mathbf{z}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(T)} \left[\frac{1}{\gamma} \left\| \hat{v}_\theta(\mathbf{z}, t) \odot m - v(\mathbf{z}_0^{\mathbf{d}}, \boldsymbol{\epsilon}, t) \odot m \right\|_2^2 \right], \tag{6}$$

where γ is the number of valid elements in m; $\hat{v}_\theta(\mathbf{z}, t)$ indicates the velocity estimated from U-Net with $\mathbf{z} = \mathrm{Cat}(\mathbf{z}^{\mathbf{x}}, \mathbf{z}^{\tilde{\mathbf{d}}'}, \mathbf{z}_0^{\mathbf{d}})$; $v(\mathbf{z}_0^{\mathbf{d}}, \boldsymbol{\epsilon}, t)$ denotes the ground-truth velocity defined as $v(\mathbf{z}_0^{\mathbf{d}}, \boldsymbol{\epsilon}, t) = \sqrt{\bar{\alpha}_t} \boldsymbol{\epsilon} - \sqrt{1 - \bar{\alpha}_t} \mathbf{z}_0^{\mathbf{d}}$ [38]. With the masked training objective, BetterDepth not only strengthens the depth conditioning by discarding significantly dissimilar patches but learns to capture fine-grained details from the remaining patch pairs without overfitting the training data.

We further analyze the effectiveness of our training strategies from the perspective of data distribution. As illustrated in Fig. 3, the learned output distribution of BetterDepth (denoted as $\hat{\mathcal{X}}$) initially covers $\mathcal{X}(\mathbf{M}_{\mathrm{DM}}, \mathbf{D}_{\mathrm{syn}})$ without either pre-alignment or patch masking, as we essentially train a diffusion model with synthetic data in Better-Depth. Thus the resulting model is able to extract fine-grained details but falls short in generalization according to Tab. 1. By applying global pre-alignment, we bring $\hat{\mathcal{X}}$ closer to the output distribution of the pre-trained depth model, *i.e.*, $\mathcal{X}(\mathbf{M}_{\mathrm{FFD}}, \{\mathbf{D}_{\mathrm{syn}}, \mathbf{D}_{\mathrm{real}}\})$, which
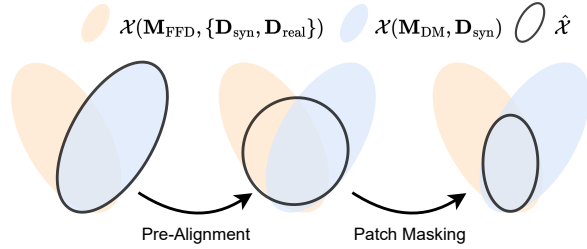

Figure 3: **Illustration of output distributions** after applying pre-alignment and patch masking. The output distribution of BetterDepth ($\hat{\mathcal{X}}$) is pushed towards the intersection of $\mathcal{X}(\mathbf{M}_{\mathrm{FFD}}, \{\mathbf{D}_{\mathrm{syn}}, \mathbf{D}_{\mathrm{real}}\})$ and $\mathcal{X}(\mathbf{M}_{\mathrm{DM}}, \mathbf{D}_{\mathrm{syn}})$ to achieve detailed zero-shot MDE.

equips BetterDepth with better zero-shot capability by enhancing the conditioning strength at the global scale. Finally, with local patch masking, we filter out significantly different patches and further shrink $\hat{\mathcal{X}}$ toward the intersection of $\mathcal{X}(\mathbf{M}_{\mathrm{FFD}}, \{\mathbf{D}_{\mathrm{syn}}, \mathbf{D}_{\mathrm{real}}\})$ and $\mathcal{X}(\mathbf{M}_{\mathrm{DM}}, \mathbf{D}_{\mathrm{syn}})$. Therefore, BetterDepth gains the advantages of both worlds and inherits the prior knowledge from the pre-trained depth model while learning to extract fine-grained details with diffusion, approximating $\mathcal{X}(\mathbf{M}_{\mathrm{ideal}}, \mathbf{D}_{\mathrm{ideal}})$ in Tab. 1.

3.4 Inference Strategies

The inference pipeline is depicted in Fig. 4. Similar to the training procedure, we first generate a coarse depth map $\tilde{\mathbf{d}}$ from the input image $\mathbf{x}$, *i.e.*, $\tilde{\mathbf{d}} = \mathbf{M}_{\mathrm{FFD}}(\mathbf{x})$, and then convert t into a latent codes $\mathbf{z}^{\mathbf{x}}$, $\mathbf{z}^{\tilde{\mathbf{d}}}$ as conditioning. In the latent space, we sample the initial value from standard Gaussian noise, *i.e.*, $\mathbf{z}_{t=T}^{\hat{\mathbf{d}}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and concatenate it with $\mathbf{z}^{\mathbf{x}}$, $\mathbf{z}^{\tilde{\mathbf{d}}}$ as input to the U-Net, $\mathbf{z} = \mathrm{Cat}(\mathbf{z}^{\mathbf{x}}, \mathbf{z}^{\tilde{\mathbf{d}}}, \mathbf{z}_t^{\hat{\mathbf{d}}})$, where the depth conditioning ensures generalization and the image conditioning provides auxiliary information for refinement. After T-step iterative refinement with the pre-trained U-Net $\hat{v}_\theta(\mathbf{z}, t)$, the clean latent $\mathbf{z}_0^{\hat{\mathbf{d}}}$ is decoded to a final depth map $\hat{\mathbf{d}}$ via the latent VAE decoder.

Plug-and-Play. Once trained, BetterDepth can directly refine the output of previously unseen MDE models, without any additional training. This advantage comes from the different roles of $\mathbf{M}_{\mathrm{FFD}}$ and $\mathbf{M}_{\mathrm{DM}}$ in BetterDepth. According to our proposed training strategy, BetterDepth treats $\mathbf{M}_{\mathrm{FFD}}$
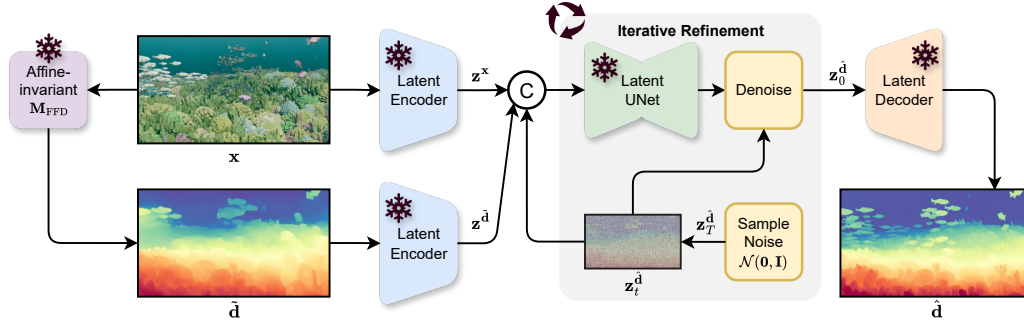
Figure 4: **BetterDepth inference pipeline.** Given an image $\mathbf{x}$ and a pre-trained depth model, we first estimate the coarse depth map $\tilde{\mathbf{d}}$ as conditioning. After converting $\mathbf{x}$ and $\tilde{\mathbf{d}}$ to latent space, we concatenate the latent codes $\mathbf{z}^{\mathbf{x}}$, $\mathbf{z}^{\tilde{\mathbf{d}}}$ with the depth latent $\mathbf{z}_t^{\hat{\mathbf{d}}}$ for denoising. After T-step refinement, random Gaussian noise $\mathbf{z}_T^{\hat{\mathbf{d}}}$ has been converted to $\mathbf{z}_0^{\hat{\mathbf{d}}}$ and is decoded to the final estimate $\hat{\mathbf{d}}$.

as the knowledge reservoir to ensure zero-shot MDE performance and utilizes $\mathbf{M}_{\mathrm{DM}}$ only to refine details. When faced with a different $\mathbf{M}_{\mathrm{FFD}}$, BetterDepth inherits a correspondingly different prior, but maintains the functionality to add fine-grained details to it. Given the increasing trend to train foundational MDE models [49], BetterDepth can be flexibly added to new models as a refinement module to enhance the extraction of details.

4 Experiments and Analysis

4.1 Experimental Settings

Implementation. We employ Depth Anything [49] as $\mathbf{M}_{\mathrm{FFD}}$ and use the Marigold architecture [17] with Stable Diffusion weight initialization [34] as $\mathbf{M}_{\mathrm{DM}}$ in our BetterDepth, where we only fine-tune the denoising U-Net. BetterDepth is trained for 5K iterations with batch size 32. The training takes around 1.5 days on a single NVIDIA RTX A6000 GPU. The Adam optimizer [18] is used with the learning rate set to 3×10^{-5}. We set the patch size $w = 8$ and the masking threshold $\eta = 0.1$ under the depth range $[-1, 1]$. For inference, we apply the DDIM scheduler with 50-step sampling [44] and obtain the final result with 10 test-time ensemble members [17].

Datasets and Evaluation. We follow Marigold [17] and use 74K samples from two synthetic datasets **Hypersim** [33] and **Virtual KITTI** [2] for training. Additionally, we construct two smaller datasets by randomly selecting 2K and 400 samples, respectively, from the full training dataset to test the performance of BetterDepth with fewer training samples (denoted as BetterDepth-2K and BetterDepth-400). For evaluation, we employ five unseen datasets **NYUv2** [28] (654 samples), **KITTI** [11] (652 samples from the Eigen test split [9]), **ETH3D** [43] (454 samples), **ScanNet** [6] (800 samples based on the Marigold split [17]), and **DIODE** [45] (325 indoor samples and 446 outdoor ones), and conduct quantitative comparisons with two metrics, AbsRel (absolute relative error: $\frac{1}{N} \sum_{k=1}^{N} |\hat{\mathbf{d}}_k - \mathbf{d}_k| / \mathbf{d}_k$ with N denoting the number of pixels) and $\delta 1$ (percentage of $\max(\mathbf{a}_i / \mathbf{d}_i, \mathbf{d}_i / \mathbf{a}_i) < 1.25$). In-the-wild images are also collected for qualitative evaluation of zero-shot MDE.

4.2 Benchmarking

In this section, we compare BetterDepth with state-of-the-art affine-invariant MDE methods to show its superior zero-shot performance and reconstruction of details.

Zero-Shot Performance. Tab. 2 shows the results for BetterDepth compared with both feed-forward and diffusion-based MDE approaches. Benefitting from the proposed framework and training strategies, BetterDepth successfully combines the geometric prior from the pre-trained depth model with the ability to model fine details. Specifically, BetterDepth-2K already achieves state-of-the-art performance and BetterDepth-400 still compares favorably to prior art. In addition, different MDE models can be directly plugged into the BetterDepth framework, which consistently improves their outputs across most datasets, as demonstrated in Tab. 3. BetterDepth also outperforms existing MDE

Table 2: **Quantitative evaluation of zero-shot performance** with state-of-the-art affine-invariant MDE methods. #Train is the amount of training data. FFD and DM correspond to feed-forward and diffusion models. Metrics are shown in percentage with best and second-best results marked. The average rank cannot be computed for DepthFM due to missing metrics on ETH3D and ScanNet.

Method	#Train	Model Type FFD	DM	NYUv2 AbsRel↓	δ1↑	KITTI AbsRel↓	δ1↑	ETH3D AbsRel↓	δ1↑	ScanNet AbsRel↓	δ1↑	DIODE AbsRel↓	δ1↑	Avg. Rank
DiverseDepth [51]	320K	✓		11.7	87.5	19.0	70.4	22.8	69.4	10.9	88.2	37.6	63.1	12.1
MiDaS [31]	2M	✓		9.5	91.5	18.3	71.1	19.0	88.4	9.9	90.7	26.6	71.3	10.3
LeReS [53]	354K	✓		9.0	91.6	14.9	78.4	17.1	77.7	9.1	91.7	27.1	76.6	9.2
Omnidata [8]	12.2M	✓		7.4	94.5	14.9	83.5	16.6	77.8	7.5	93.6	33.9	74.2	8.9
HDN [55]	300K	✓		6.9	94.8	11.5	86.7	12.1	83.3	8.0	93.9	24.6	78.0	6.9
DPT [30]	1.4M	✓		9.1	91.9	11.1	88.1	11.5	92.9	8.4	93.2	26.9	73.0	8.3
Depth Anything [49]	63.5M	✓		4.3	98.0	8.0	94.6	6.2	98.0	4.3	98.1	26.0	75.9	2.9
Marigold [17]	74K		✓	5.5	96.4	9.9	91.6	6.5	96.0	6.4	95.1	30.8	77.3	5.6
DepthFM [12]	63K		✓	6.5	95.6	8.3	93.4	-	-	-	-	22.5	80.0	-
GeoWizard [10]	280K		✓	5.2	96.6	9.7	92.1	6.4	96.1	6.1	95.3	29.7	79.2	5.2
BetterDepth-400 (Ours)	400	✓	✓	4.6	97.9	7.9	94.5	5.0	97.8	4.6	97.8	21.9	75.3	4.0
BetterDepth-2K (Ours)	2K	✓	✓	4.4	97.9	7.4	95.1	4.7	98.1	4.3	98.0	22.0	75.5	2.7
BetterDepth (Ours)	74K	✓	✓	4.2	98.0	7.5	95.2	4.7	98.1	4.3	98.1	22.6	75.5	1.8

Table 3: **Plug-and-play experiments.** BetterDepth directly works with CNN-based (MiDaS [31]) and transformer-based MDE models (DPT [30]), improving their results without re-training.

Method	NYUv2 AbsRel↓	δ1↑	KITTI AbsRel↓	δ1↑	ETH3D AbsRel↓	δ1↑	ScanNet AbsRel↓	δ1↑	DIODE AbsRel↓	δ1↑
MiDaS [31]	9.5	91.5	18.3	71.1	19.0	88.4	9.9	90.7	26.6	71.3
BetterDepth+MiDaS	8.4	93.4	15.1	78.4	17.9	91.2	9.3	91.6	26.6	71.9
Improvement	1.1	1.9	3.2	7.3	1.1	2.8	0.6	0.9	0.0	0.3
DPT [30]	9.1	91.9	11.1	88.1	11.5	92.9	8.4	93.2	26.9	73.0
BetterDepth+DPT	7.9	93.7	10.0	89.8	10.3	94.5	7.8	93.8	26.5	73.6
Improvement	1.2	1.8	1.1	1.7	1.2	1.6	0.6	0.6	0.4	0.6

Table 4: **Quantitative evaluation of detail extraction** on Middlebury 2014 [41]. Edge-based metrics, *i.e.*, the completeness and accuracy of depth boundaries (DBE_comp and DBE_acc) [20] and the edge precision and recall (EP and ER) [14], are also shown to evaluate performance specifically on high-frequency details. The best and second-best results are marked.

Method	AbsRel (%) ↓	δ1 (%) ↑	DBE_comp ↓	DBE_acc ↓	EP (%) ↑	ER (%) ↑
Marigold [17]	7.57	93.24	5.60	3.09	16.65	23.75
Depth Anything [49]	3.14	99.44	6.35	2.66	24.73	16.12
BetterDepth (Ours)	2.95	99.52	3.61	2.09	28.49	50.35

approaches in visual quality as depicted in Fig. 1 and 5. Compared with previous methods that either suffer from over-smoothing or inaccurate depth layout, BetterDepth correctly recovers the spatial structure of different scenes while capturing small details, leading to visually improved results.

Fine-Grained Detail Extraction. Despite achieving state-of-the-art performance, Tab. 2 cannot fully represent the performance of BetterDepth (especially w.r.t. details), as the depth labels in commonly used datasets are sparse or noisy, *e.g.*, Fig. A8-A17. Thus, we further evaluate the ability to reconstruct details on a high-resolution RGB-D dataset Middlebury 2014 [41]. Four additional edge-based metrics are employed to focus on depth discontinuities: the completeness and accuracy of depth boundary errors [20] and the precision and recall for edges [14]. As shown in Tab. 4, BetterDepth delivers more accurate estimates in terms of both global and edge-based metrics and succeeds in capturing challenging details, *e.g.*, the fine mesh in Fig. 6.

4.3 Ablation Study

In Tab. 5, we study the effectiveness of each design choice in BetterDepth and draw the following conclusions: (i) **Depth Conditioning.** Without depth conditioning, model #1 in Tab. 5 performs
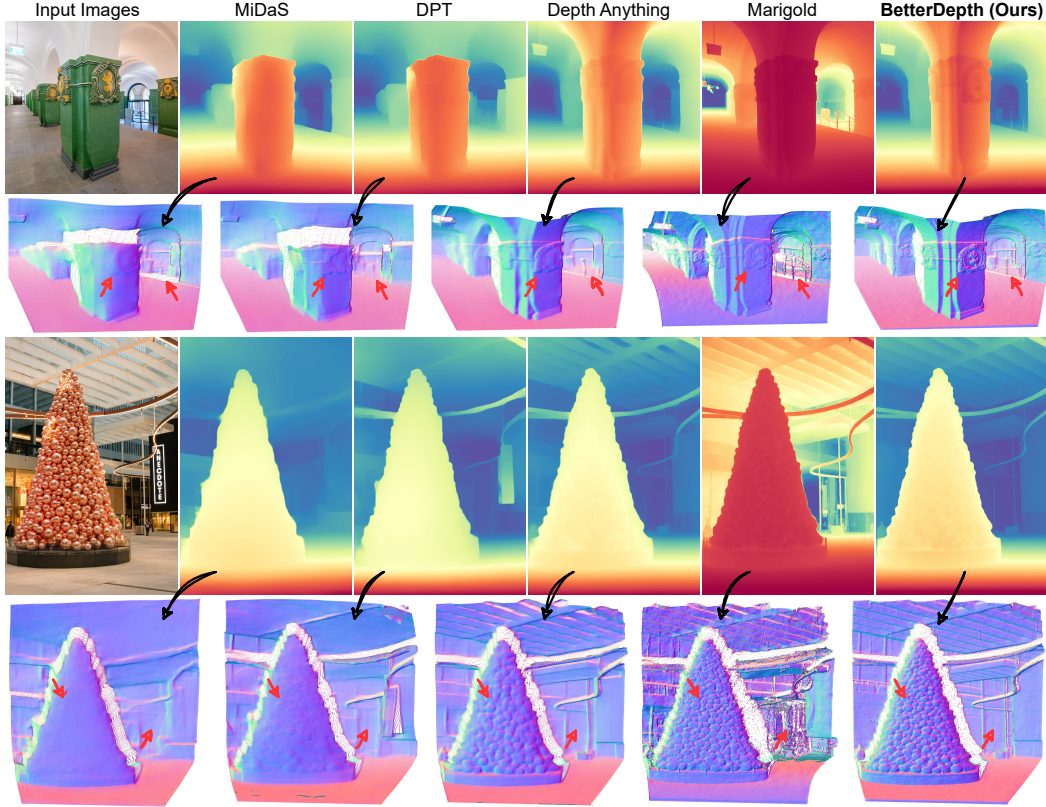
Figure 5: **Qualitative comparisons** of depth estimation and 3D reconstruction results (colored as normals), where Marigold predicts depth values and the others output disparity.

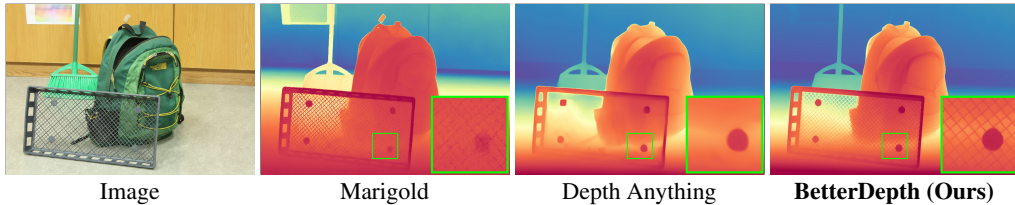| Image | Marigold | Depth Anything | **BetterDepth (Ours)** |

Figure 6: **Visual comparisons** on Middlebury 2014 [41]. Details are zoomed in.

Table 5: **Ablation study**. All variants are trained on the full 74K training pairs for 5K iterations. The best and second-best results are marked.

ID	Depth Conditioning	Global Pre-Alignment	Local Patch Masking	NYUv2 AbsRel↓	δ1↑	KITTI AbsRel↓	δ1↑
#1	✗	✗	✗	6.1	96.1	9.1	90.7
#2	✓	✗	✗	5.2	97.0	8.6	92.2
#3	✓	✓	✗	4.7	97.5	7.9	94.4
#4	✓	✓	✓	4.2	98.0	7.5	95.2

similarly to previous diffusion-based methods like Marigold [17], and struggles with generalization only from synthetic training data. By utilizing the geometric prior from the pre-trained depth estimator, model #2 achieves consistent improvements in both indoor and outdoor scenarios, as shown in Tab. 5. (ii) **Global Pre-Alignment.** Despite the improvements gained with depth conditioning, we find the zero-shot performance still remains below the pre-trained Depth Anything model [49]. In other words, even having good depth maps from the pre-trained model as initialization, the naive
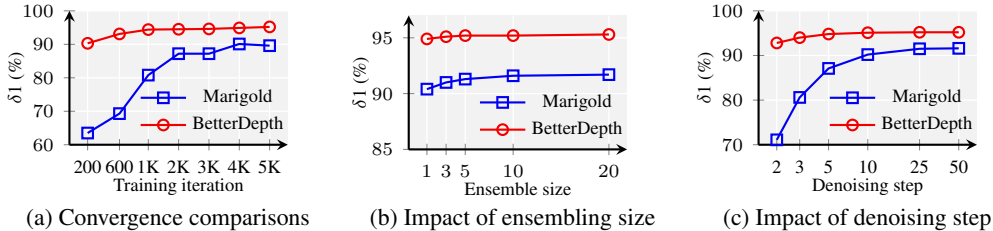
9

Figure 7: **Training and inference efficiency** compared with Marigold [17] on the KITTI dataset.

conditioning model (#2) struggles to balance the contribution of different priors and does not yield an improvement. This is because model #2 overfits the distribution of training data and under-utilizes the prior knowledge in the pre-trained model. By aligning the depth conditioning to the ground truth during training, model #3 better learns to follow the depth conditioning at a global scale and brings further improvements in zero-shot generalization. (iii) **Local Patch Masking.** Our full model #4, with the masked training objective, exhibits the best performance. By filtering out significantly dissimilar regions with patch masking, we ensure that BetterDepth closely adheres to depth conditioning at local scales, thus better exploiting the prior for zero-shot transfer. Meanwhile, operating at patch level fully retains the information in local regions and thus benefits the reconstruction of details, *e.g.*, edges and fine structures, as illustrated in Fig. 1 and 5.

4.4 Method Analysis

In this section, we further analyze BetterDepth with respect to training and inference efficiency.

Training Efficiency. We compare the training efficiency of BetterDepth with the state-of-the-art diffusion-based method Marigold [17]. Helped by the additional depth conditioning, BetterDepth converges significantly faster than Marigold, as depicted in Fig. 7a. With only 200 iterations (≈ 1.5 hours of training), BetterDepth achieves comparable performance to Marigold trained with 5K iterations. Furthermore, since we must only learn to refine details, thanks to the proposed training strategies, BetterDepth outperforms Marigold with fewer training samples, *e.g.*, BetterDepth-400 in Tab. 2, validating the overall strategy.

Inference Efficiency. We compare the efficiency at inference time with different ensemble sizes and numbers of denoising steps. Test-time ensembling aims to aggregate information from multiple predictions, and larger ensemble sizes generally bring better and more stable results [17]. As depicted in Fig. 7b, on KITTI the $\delta 1$ difference between a single inference and an ensemble of 10 members is 1.2 percentage points for Marigold but only 0.4 for BetterDepth, confirming its better stability. Meanwhile, BetterDepth produces comparable or even better results than 50-step Marigold with only 2 inference steps, as shown in Fig. 7c. In terms of inference speed, the 50-step Marigold achieves 91.6% $\delta 1$ accuracy on KITTI with 10 ensemble members, spending 30.5 seconds per sample on an NVIDIA GeForce RTX 4090 GPU. In contrast, our 2-step BetterDepth achieves 92.5% $\delta 1$ accuracy in a single inference pass with only 0.4 seconds per sample (0.38 seconds for the diffusion denoising and 0.02 seconds for the depth conditioning prediction).

5 Conclusion

We have presented BetterDepth to achieve robust, detailed, and efficient affine-invariant monocular depth estimates. The proposed method combines the strong prior of massively pre-trained MDE models with the recovery of fine details enabled by diffusion models, and devises training strategies to maximally retain the strengths of both discriminative depth estimation and conditional depth map generation. In this way, BetterDepth achieves state-of-the-art MDE performance and is able to refine different feed-forward depth estimators without re-training.

References

[1] Amir Atapour-Abarghouei and Toby P Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *CVPR*, pages 2800–2810, 2018.

[2] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv preprint arXiv:2001.10773*, 2020.

[3] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In *NIPS*, volume 29, 2016.

[4] Weifeng Chen, Shengyi Qian, David Fan, Noriyuki Kojima, Max Hamilton, and Jia Deng. Oasis: A large-scale dataset for single image 3d in the wild. In *CVPR*, pages 679–688, 2020.

[5] Zheng Chen, Yulun Zhang, Ding Liu, Jinjin Gu, Linghe Kong, Xin Yuan, et al. Hierarchical integration diffusion model for realistic image deblurring. In *NIPS*, volume 36, 2023.

[6] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

[7] Yiqun Duan, Xianda Guo, and Zheng Zhu. Diffusiondepth: Diffusion denoising approach for monocular depth estimation. *arXiv preprint arXiv:2303.05021*, 2023.

[8] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *ICCV*, pages 10786–10796, 2021.

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[10] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. *arXiv preprint arXiv:2403.12013*, 2024.

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

[12] Ming Gui, Johannes S Fischer, Ulrich Prestel, Pingchuan Ma, Dmytro Kotovenko, Olga Grebenkova, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Depthfm: Fast monocular depth estimation with flow matching. *arXiv preprint arXiv:2403.13788*, 2024.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NIPS*, volume 33, pages 6840–6851, 2020.

[14] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *WACV*, pages 1043–1051. IEEE, 2019.

[15] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *PAMI*, 2024.

[16] Yuanfeng Ji, Zhe Chen, Enze Xie, Lanqing Hong, Xihui Liu, Zhaoqiang Liu, Tong Lu, Zhenguo Li, and Ping Luo. Ddp: Diffusion model for dense visual prediction. In *ICCV*, pages 21741–21752, 2023.

[17] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, 2024.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[20] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Korner. Evaluation of cnn-based single-image depth estimation methods. In *ECCVW*, 2018.

[21] Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun. A survey on deep learning techniques for stereo-based depth estimation. *PAMI*, 44(4):1738–1764, 2020.

[22] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022.

[23] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018.

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021.

[25] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, pages 11461–11471, 2022.

[26] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

[27] Lukas Mehl, Andrés Bruhn, Markus Gross, and Christopher Schroers. Stereo conversion with disparity-aware warping, compositing and inpainting. In *WACV*, pages 4260–4269, 2024.

[28] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012.

[29] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *CVPR*, pages 10106–10116, 2024.

[30] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, pages 12179–12188, 2021.

[31] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *PAMI*, 44(3):1623–1637, 2020.

[32] Lucas Relic, Roberto Azevedo, Markus Gross, and Christopher Schroers. Lossy image compression with foundation diffusion models. *arXiv preprint arXiv:2404.08580*, 2024.

[33] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021.

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[35] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH*, pages 1–10, 2022.

[36] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NIPS*, volume 35, pages 36479–36494, 2022.

[37] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *PAMI*, 45(4):4713–4726, 2022.

[38] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

[39] Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. In *NIPS*, volume 36, 2023.

[40] Saurabh Saxena, Abhishek Kar, Mohammad Norouzi, and David J Fleet. Monocular depth estimation using diffusion models. *arXiv preprint arXiv:2302.14816*, 2023.

[41] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *PR*, pages 31–42. Springer, 2014.

[42] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR*, volume 1, pages I–I. IEEE, 2003.

[43] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, pages 3260–3269, 2017.

[44] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[45] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *arXiv preprint arXiv:1908.00463*, 2019.

[46] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, pages 8445–8453, 2019.

[47] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *CVPR*, pages 16293–16303, 2022.

[48] Diana Wofk, Fangchang Ma, Tien-Ju Yang, Sertac Karaman, and Vivienne Sze. Fastdepth: Fast monocular depth estimation on embedded systems. In *ICRA*, pages 6101–6108. IEEE, 2019.

[49] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[50] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *NIPS*, 2024.

[51] Wei Yin, Xinlong Wang, Chunhua Shen, Yifan Liu, Zhi Tian, Songcen Xu, Changming Sun, and Dou Renyin. Diversedepth: Affine-invariant depth prediction using diverse data. *arXiv preprint arXiv:2002.00569*, 2020.

[52] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *ICCV*, pages 9043–9053, 2023.

[53] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *CVPR*, pages 204–213, 2021.

[54] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. In *ICLR*, 2020.

[55] Chi Zhang, Wei Yin, Billzb Wang, Gang Yu, Bin Fu, and Chunhua Shen. Hierarchical normalization for robust monocular depth estimation. In *NIPS*, volume 35, pages 14128–14139, 2022.

[56] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *ICCV*, pages 5729–5739, 2023.

Appendix

In this appendix, we provide more implementation details, experiments, analysis, and discussions for a comprehensive evaluation and understanding of BetterDepth. Detailed contents are listed as follows:

Contents of Appendix

Algorithm 1 BetterDepth Training Procedure

1: **repeat**
2:   $(\mathbf{x}, \mathbf{d}) \sim \mathbf{D}_{\mathrm{syn}}$ ▷ Sample image and depth label
3:   $\tilde{\mathbf{d}} = \mathbf{M}_{\mathrm{FFD}}(\mathbf{x})$ ▷ Estimate coarse depth as conditioning
4:   $\tilde{\mathbf{d}}' = s\tilde{\mathbf{d}} + b$ with $(s, b) = \arg\min\limits_{s,b} \left\| s\tilde{\mathbf{d}} + b - \mathbf{d} \right\|_2^2,$ ▷ Global pre-alignment
5:   $m = \mathrm{PatchMaskEstimate}(\tilde{\mathbf{d}}', \mathbf{d})$ ▷ Estimate patch mask
6:   $\mathbf{z}^{\mathbf{x}} = \mathcal{E}(\mathbf{x}), \mathbf{z}^{\tilde{\mathbf{d}}'} = \mathcal{E}(\tilde{\mathbf{d}}'), \mathbf{z}^{\mathbf{d}} = \mathcal{E}(\mathbf{d})$ ▷ Encode with frozen latent encoder $\mathcal{E}$
7:   $t \sim \mathrm{Uniform}(\{1, \ldots, T\}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Sample timestamp and Gaussian noise
8:   $\mathbf{z}_t^{\mathbf{d}} = \sqrt{\bar{\alpha}_t}\mathbf{z}^{\mathbf{d}} + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}$ ▷ Add noise with velocity prediction method
9:   $\mathbf{z} = \mathrm{Cat}(\mathbf{z}^{\mathbf{x}}, \mathbf{z}^{\tilde{\mathbf{d}}'}, \mathbf{z}_t^{\mathbf{d}})$ ▷ Concatenate latent features as U-Net input
10: $v(\mathbf{z}^{\mathbf{d}}, \boldsymbol{\epsilon}, t) = \sqrt{\bar{\alpha}_t}\boldsymbol{\epsilon} - \sqrt{1 - \bar{\alpha}_t}\mathbf{z}^{\mathbf{d}}$ ▷ Compute ground-truth velocity
11: Take gradient descent step on
        $\nabla_\theta \frac{1}{\gamma} \left\| \hat{v}_\theta(\mathbf{z}, t) \odot m - v(\mathbf{z}^{\mathbf{d}}, \boldsymbol{\epsilon}, t) \odot m \right\|_2^2$ ▷ Train latent U-Net with masked objective
12: **until** converged

A   Training Procedure

Algorithm 1 displays the complete training procedure for the proposed BetterDepth method, where the output type of BetterDepth is consistent with that of the employed $\mathbf{M}_{\mathrm{FFD}}$, *e.g.*, our BetterDepth predicts affine-invariant inverse depth as Depth Anything [49]. Compared with the previous diffusion training scheme for MDE models [39, 17, 10], we first design a depth-conditioned framework to efficiently utilize the rich geometric prior from pre-trained depth models. In addition, global pre-alignment and local patch masking methods are proposed to enable learning detail refinement while maintaining the faithfulness of BetterDepth to depth conditioning, achieving robust zero-shot MDE performance with fine-grained details.
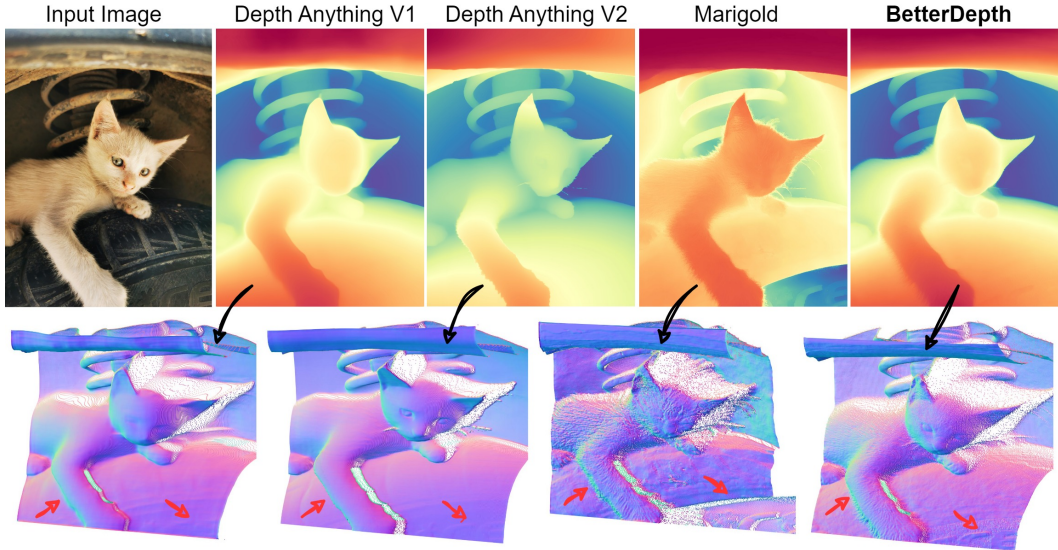
Figure A1: Visual comparisons with Depth Anything V2 [50].

Table A1: **Quantitative evaluation of zero-shot performance** on five unseen datasets. The best and second-best results are marked.

Method	NYUv2		KITTI		ETH3D		ScanNet		DIODE		Avg.
	AbsRel↓	δ1↑	AbsRel↓	δ1↑	AbsRel↓	δ1↑	AbsRel↓	δ1↑	AbsRel↓	δ1↑	Rank
Marigold [17]	5.5	96.4	9.9	91.6	6.5	96.0	6.4	95.1	30.8	77.3	3.7
Depth Anything [49]	4.3	98.0	8.0	94.6	6.2	98.0	4.3	98.1	26.0	75.9	1.9
Depth Anything V2 [50]	4.4	97.8	8.3	93.9	6.2	98.2	4.2	97.8	26.4	75.4	2.6
BetterDepth (Ours)	4.2	98.0	7.5	95.2	4.7	98.1	4.3	98.1	22.6	75.5	1.4

Table A2: **Quantitative evaluation of detail extraction performance** on the high-resolution dataset Middlebury 2014 [41]. Edge-based metrics, *i.e.*, the completeness and accuracy of depth boundary errors (denoted as DBE_comp and DBE_acc) [20] and the edge precision and edge recall (denoted as EP and ER) [14], are also employed to evaluate detail extraction performance. The best and second-best results are marked.

Method	AbsRel (%) ↓	δ1 (%) ↑	DBE_comp ↓	DBE_acc ↓	EP (%) ↑	ER (%) ↑	Avg. Rank
Marigold [17]	7.57	93.24	5.60	3.09	16.65	23.75	3.7
Depth Anything [49]	3.14	99.44	6.35	2.66	24.73	16.12	3.2
Depth Anything V2 [50]	3.06	99.38	4.19	2.23	26.74	35.89	2.2
BetterDepth (Ours)	2.95	99.52	3.61	2.09	28.49	50.35	1

B Comparison with Depth Anything V2

In this section, we compare BetterDepth to the concurrent work Depth Anything V2 [50]. By training on high-quality synthetic datasets, Depth Anything V2 achieves significant performance improvements, *e.g.*, fine detail and transparent objects, over Depth Anything [49]. However, we found that both the **training dataset** and the **model architecture** are crucial for MDE performance. As shown in Tab. A1 and A2, although Depth Anything V2 achieves promising performance in detail extraction, our BetterDepth still exhibits better performance even with much less synthetic training data (595K in Depth Anything V2 v.s. 74K in BetterDepth), thanks to the iterative refinement of diffusion model. In addition, BetterDepth also captures better details like the cat's hair in Fig. A1, validating its overall best performance.

C Combination of Prior Knowledge

Due to the ill-posedness of the MDE task, rich prior knowledge has been proven important in accurate depth estimation from single-view input [31, 30, 17, 49]. Unlike previous MDE methods that mainly exploit single-sourced knowledge, *e.g.*, geometric priors in MiDaS [31] or image priors in Marigold [17], our BetterDepth

Table A3: **Contribution of the geometric prior and the image prior in BetterDepth**, where geometric and image priors correspond to the knowledge gained from the pre-trained depth model, *i.e.*, Depth Anything [49], and the Stable Diffusion model [34]. The model without geometric prior uses the same network and fine-tuning method as Marigold [17] but estimates inverse depth (following Depth Anything [49]) instead of relative depth. For the model without image prior, we follow Stable Diffusion [34] to train the latent UNet from scratch and keep the pre-trained VAE unchanged. Metrics are shown in percentage terms, where the best and second-best results are marked.

Geometric Prior	Image Prior	NYUv2		KITTI		ETH3D		ScanNet		DIODE	
		AbsRel↓	δ1↑	AbsRel↓	δ1↑	AbsRel↓	δ1↑	AbsRel↓	δ1↑	AbsRel↓	δ1↑
	✓	6.1	96.1	9.1	90.7	8.5	96.1	6.5	95.0	22.2	73.7
✓		4.3	98.0	8.0	94.4	5.5	97.8	4.4	98.1	22.6	75.1
✓	✓	4.2	98.0	7.5	95.2	4.7	98.1	4.3	98.1	22.6	75.5

Table A4: **Performance of the BetterDepth trained with DPT [30].** * means the previously unseen models, *i.e.*, MiDaS [31] and Depth Anything [49], are directly plugged into the BetterDepth framework (pre-trained with DPT) for improved MDE performance. The best and second-best results are marked.

Method	NYUv2		KITTI		ETH3D		ScanNet		DIODE		Avg.
	AbsRel↓	δ1↑	AbsRel↓	δ1↑	AbsRel↓	δ1↑	AbsRel↓	δ1↑	AbsRel↓	δ1↑	Rank
MiDaS [31]	9.5	91.5	18.3	71.1	19.0	88.4	9.9	90.7	26.6	71.3	5.7
DPT [30]	9.1	91.9	11.1	88.1	11.5	92.9	8.4	93.2	26.9	73.0	4.1
Depth Anything [49]	4.3	98.0	8.0	94.6	6.2	98.0	4.3	98.1	26.0	75.9	1.5
BetterDepth+MiDaS*	7.7	94.3	13.5	81.9	17.8	92.5	8.8	92.3	26.9	72.0	4.7
BetterDepth+DPT	7.3	94.5	9.9	90.4	11.9	95.1	7.5	94.3	27.2	73.6	3.4
BetterDepth+Depth Anything*	4.3	98.1	7.9	94.7	5.5	97.9	4.3	98.1	23.0	75.3	1.2

combines knowledge from different domains. Specifically, BetterDepth utilizes the geometric prior from the pre-trained MDE models, which contains task-specific knowledge for robust depth estimation. Furthermore, BetterDepth also exploits the rich image prior via the Stable Diffusion weight initialization [34], benefiting the extraction of fine-grained details. To investigate the contribution of geometric and image priors in BetterDepth, a related ablation experiment is performed in Tab. A3. It is evident that combining prior knowledge from different sources leads to the best MDE performance.

D   More BetterDepth Variants

Apart from the BetterDepth model trained with Depth Anything [49], we additionally train a BetterDepth variant in combination with DPT [30] to further verify the effectiveness and flexibility of our proposed method. As demonstrated in Tab. A4, BetterDepth+DPT achieves 0.65/1.76% average performance gain over DPT on AbsRel/δ1 accuracy across all datasets. When directly combined with previously unseen MDE models, *i.e.*, MiDaS [31] and Depth Anything [49], BetterDepth also demonstrates general improvements on public zero-shot datasets, showing the flexibility of our proposed method in practical usage.
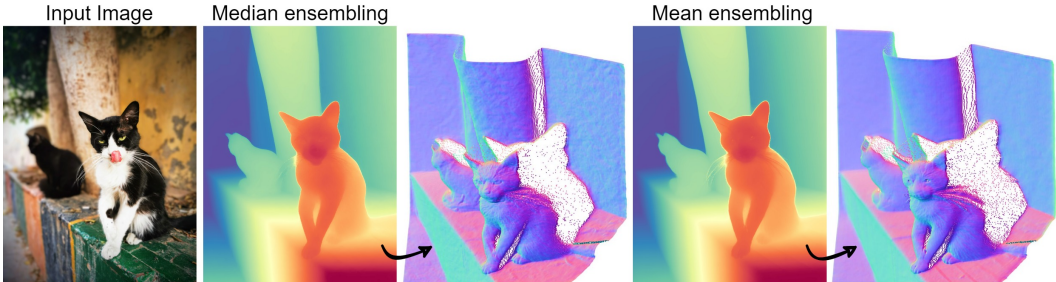

Figure A2: BetterDepth results, where mean ensembling alleviates the wobble effects.

Table A5: **Performance of BetterDepth with different test-time ensembling methods**. The best results are marked.

Method	NYUv2		KITTI		ETH3D		ScanNet		DIODE	
	AbsRel↓	δ1↑	AbsRel↓	δ1↑	AbsRel↓	δ1↑	AbsRel↓	δ1↑	AbsRel↓	δ1↑
Median ensembling	4.2	98.0	7.5	95.2	4.7	98.1	4.3	98.1	22.6	75.5
Mean ensembling	4.2	98.1	7.4	95.3	4.6	98.1	4.3	98.1	22.5	75.5

E   Noise Suppression with Mean Ensembling

Due to the random noise in the diffusion process, diffusion-based MDE methods, *e.g.*, Marigold and BetterDepth, tend to introduce subtle variations in the results, like the wobble effects in the surface normals shown in Fig. 1. A simple fix to this issue is to replace the default median operation in the test-time ensembling [17] with the mean operation, which smooths the estimated depth with multiple predictions. As shown in Fig. A2 and Tab. A5, the mean ensembling approach alleviates the wobble effects and achieves slightly better depth estimation.

F   Hyperparameter Analysis

F.1   Influence of Patch Size

Patch size w is a hyper-parameter used to estimate patch masks for training. To investigate its impact on monocular depth estimation (MDE) performance, we conduct experiments with different choices of w from 8 to 128, where 8 is the minimal patch size as the employed VAE latent encoder performs $8\times$ downscaling for pixel-to-latent conver-

Figure A3: **Influence of patch size** on NYUv2 and KITTI.

sion. As depicted in Fig. A3, the overall MDE performance fluctuates with different patch sizes, and we find setting $w = 8$ leads to the overall best performance, indicating that small patches are sufficient for learning detail refinement.
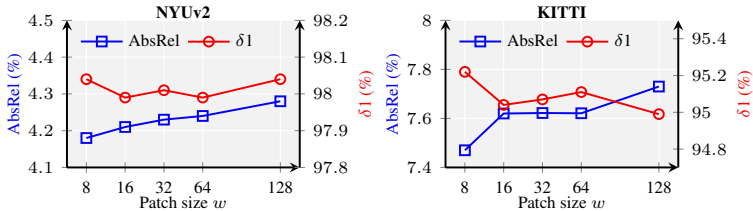
F.2   Masking Threshold and Trade-Off

The masking threshold η determines the difference tolerance level between local patches $\{\tilde{\mathbf{d}}'_n\}$ and $\{\mathbf{d}_n\}$ to filter significantly dissimilar regions during training. Since inputs are all converted to $[-1, 1]$ space before feeding into the VAE latent encoder, we conduct experiments with η varying from 0.05 to 0.30, as shown in Fig. A4. Lower η gener-

Figure A4: **Influence of masking threshold** on NYUv2 and KITTI.

ally means stricter filtering, *i.e.*, the remaining patch pairs $\tilde{\mathbf{d}}'_n$ and $\mathbf{d}_n$ are more similar to each other, and thus often leads to stronger conditioning strength. By contrast, higher η is more tolerant when selecting patches and leaves more room for learning detail refinement. Thus, the hyperparameter η controls the trade-off between depth conditioning strength and detail refinement performance, and we find a sweet spot at $\eta = 0.1$, which shows a good balance in both aspects and achieves the overall best MDE results.
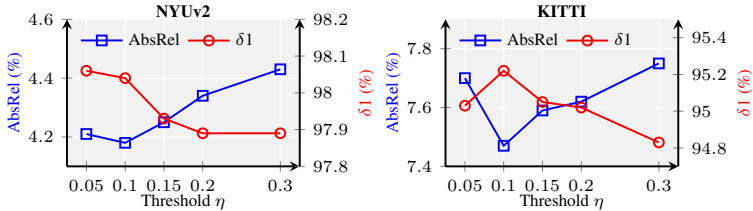
G   Error Bar Analysis

Due to the stochastic nature of diffusion models, we perform error bar analysis to evaluate the performance stability of BetterDepth on the NYUv2 dataset [28]. Instead of employing the test-time ensembling technique [17], we directly generate 10 predictions for the same input with 50 denoising steps and then compute the metrics for each estimate. Finally, we obtain the mean and standard deviation on the NYUv2 dataset and compare them with the state-of-the-art diffusion-based MDE method Marigold [17] under the same setting. As illustrated in Fig. A5, BetterDepth shows

Figure A5: **Error bar analysis** on NYUv2.

significantly better results on both AbsRel and $\delta 1$ accuracy metrics than Marigold. Meanwhile, thanks to the informative geometric cues embedded in the depth conditioning, BetterDepth also exhibits more stable MDE performance than Marigold.
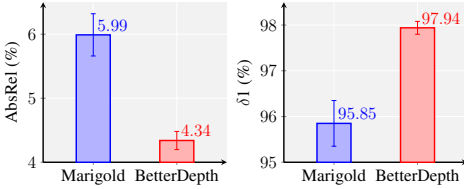
H   More Visual Results

We provide more visual comparisons on both in-the-wild scenes (Fig. A6 and A7) and public datasets (Fig. A8-A17). In-the-wild images are captured on diverse indoor/outdoor scenes with varying camera perspectives. The 3D reconstruction results colored with surface normals are also provided in Fig. A6 and A7 for better comparison of detail extraction. By contrast, public datasets contain more specific scenarios, *e.g.*, the indoor dataset NYUv2 [28] and the driving-scene dataset KITTI [11]. Overall, the proposed BetterDepth shows the best performance in estimating the accurate layout of target scenes and extracting fine-grained scene details.

I   Limitation and Future Work

While remarkable performance is achieved by BetterDepth, limitations still exist: (i) **Model Size and Inference Speed.** Since BetterDepth comprises a pre-trained MDE model and a diffusion-based refiner, the model size is determined by the chosen architectures of both components. Apart from focusing on the utilization of large foundation models, we also plan to investigate the possibility of using more lightweight components in the BetterDepth framework, *e.g.*, efficient U-Net [36] as the diffusion refiner, in future research to benefit efficient deployment in practice. In addition, the inference speed is also bounded by the chosen depth model and diffusion network, where the diffusion part usually poses the trade-off between speed and quality [13, 17]. Although BetterDepth could potentially boost speed using fewer ensemble members and fewer denoising steps, with slight performance drops as depicted in Fig. 7b and 7c, techniques like latent consistency models [26] could also be taken into account for further improvements. (ii) **Utilization of Training Data**. From the perspective of the training strategies, better pre-alignment approaches like outlier-aware methods could have more patches survive during training for better performance. Although the models trained with small datasets, *e.g.*, BetterDepth-2K in Tab. 2, already achieve comparable results to our full model, indicating that limited patches can be sufficient, better alignment methods could potentially improve patch preservation to further boost the training. (iii) **Metric Depth**. Finally, improving the performance of metric depth estimation [52, 15, 29] and transferring affine-invariant depth to metric depth are promising directions but pose several challenges, *e.g.*, scale/shift ambiguity and diverse depth ranges. It would be interesting to unlock the potential of BetterDepth in metric depth estimation, and we leave it as future work.

J   Discussion of Societal Impacts

Our work aims to improve the depth estimation performance from a single image with a similar scope to other MDE methods. BetterDepth represents progress towards zero-shot, highly detailed depth estimation, and thus it might amplify any impacts that MDE has in the societal context. On the one hand, because of the flexibility of extracting depth information from a single image, MDE can potentially benefit a variety of real-world applications, including autonomous driving [46, 54], robotics [48], and film production [27]. With the improved performance, BetterDepth could bring positive societal impacts such as providing more realistic 3D models, enhancing the precision of depth perception in autonomous vehicles, and accelerating the stereo conversion process for 3D movies. On the other hand, MDE could, like many other computer vision techniques, have negative societal impacts when used improperly. For instance, depth estimation in surveillance systems might raise privacy concerns since it can potentially enable more invasive monitoring and tracking of individuals in public spaces.
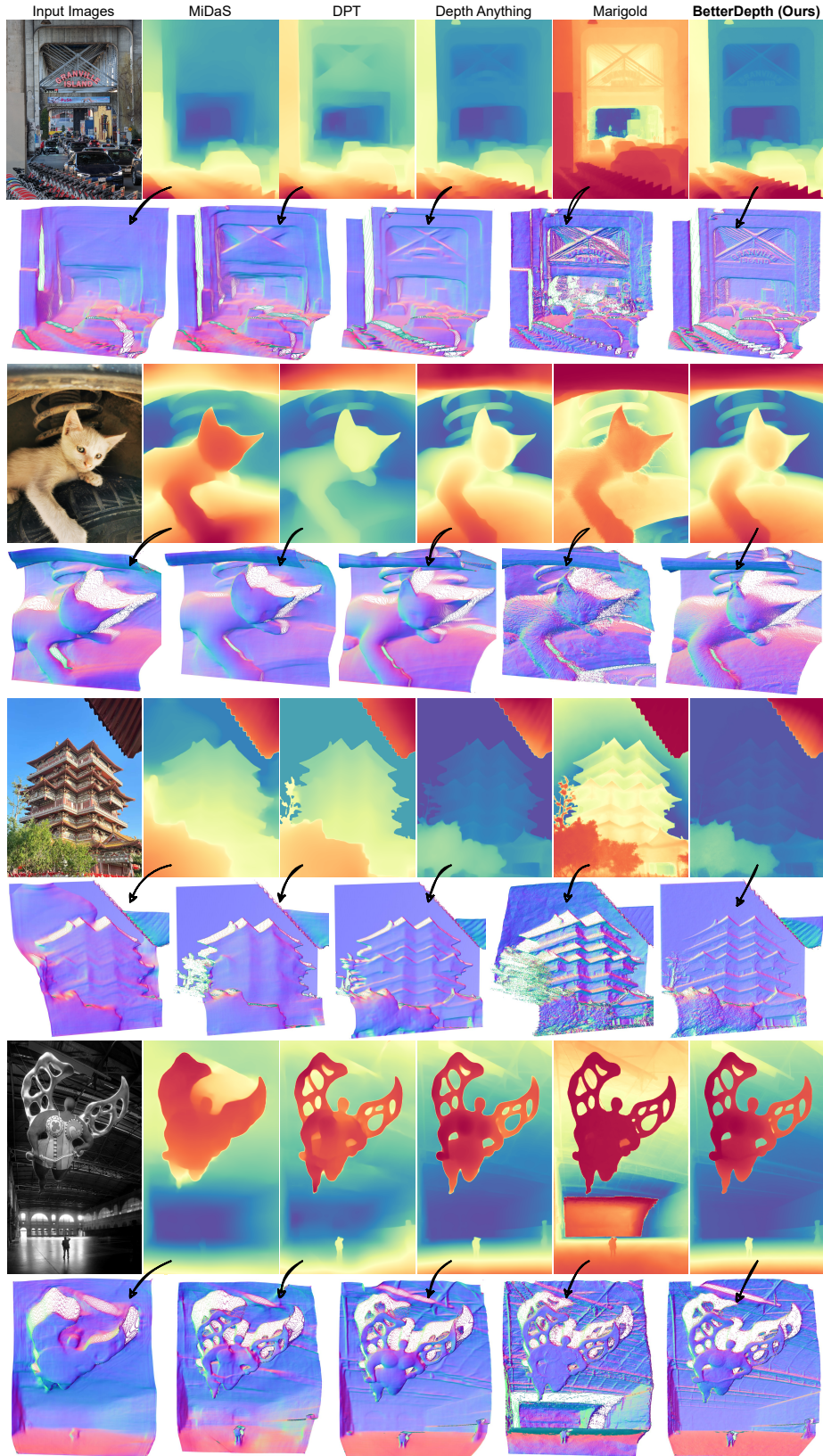
Figure A6: **Qualitative comparisons on in-the-wild samples,** part 1. Marigold predicts depth while the others output disparity values. Red indicates the close plane and blue means the far plane.
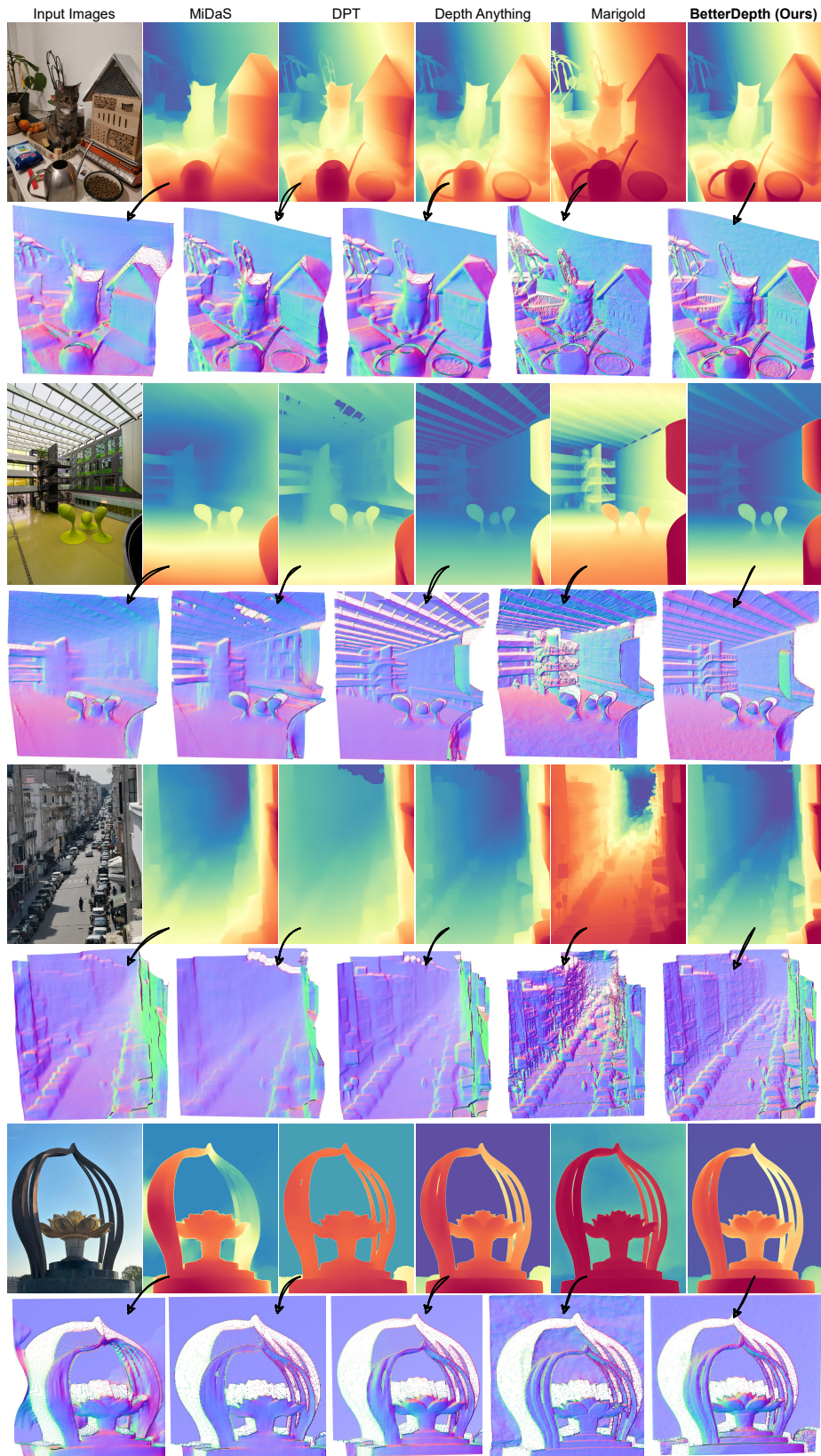
Figure A7: **Qualitative comparisons on in-the-wild samples,** part 2. Marigold predicts depth while the others output disparity values. Red indicates the close plane and blue means the far plane.

| Input Image | DPT [31] | Depth Anything [49] |
| Ground Truth | Marigold [17] | **BetterDepth (Ours)** |

| Input Image | DPT [31] | Depth Anything [49] |
| Ground Truth | Marigold [17] | **BetterDepth (Ours)** |

| Input Image | DPT [31] | Depth Anything [49] |
| Ground Truth | Marigold [17] | **BetterDepth (Ours)** |

Figure A8: **Qualitative comparisons on the NYUv2 dataset [28]**, part 1. Predictions are aligned to ground truth. For better visualization, color coding is consistent across all results, where red indicates the close plane and blue means the far plane.

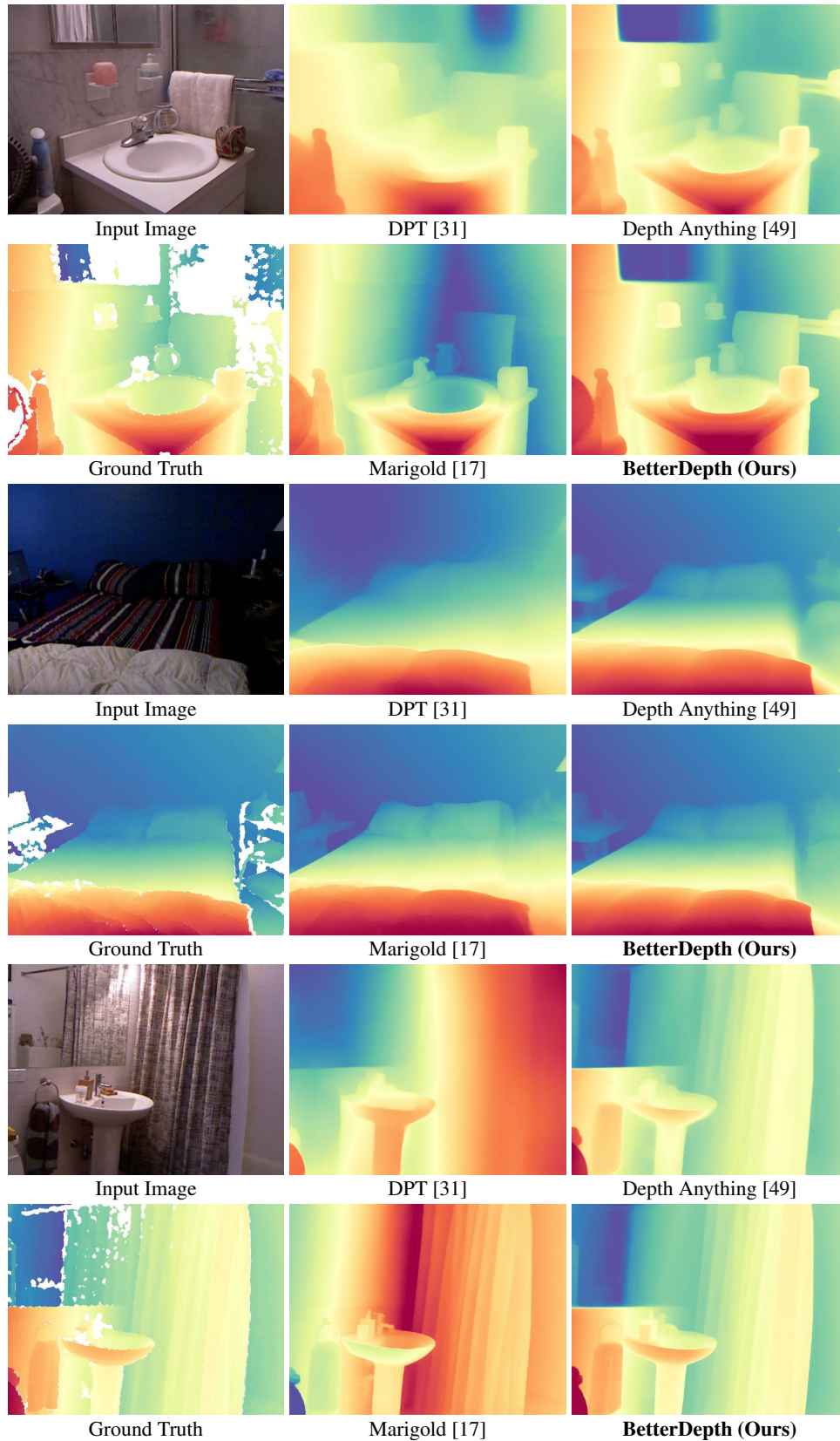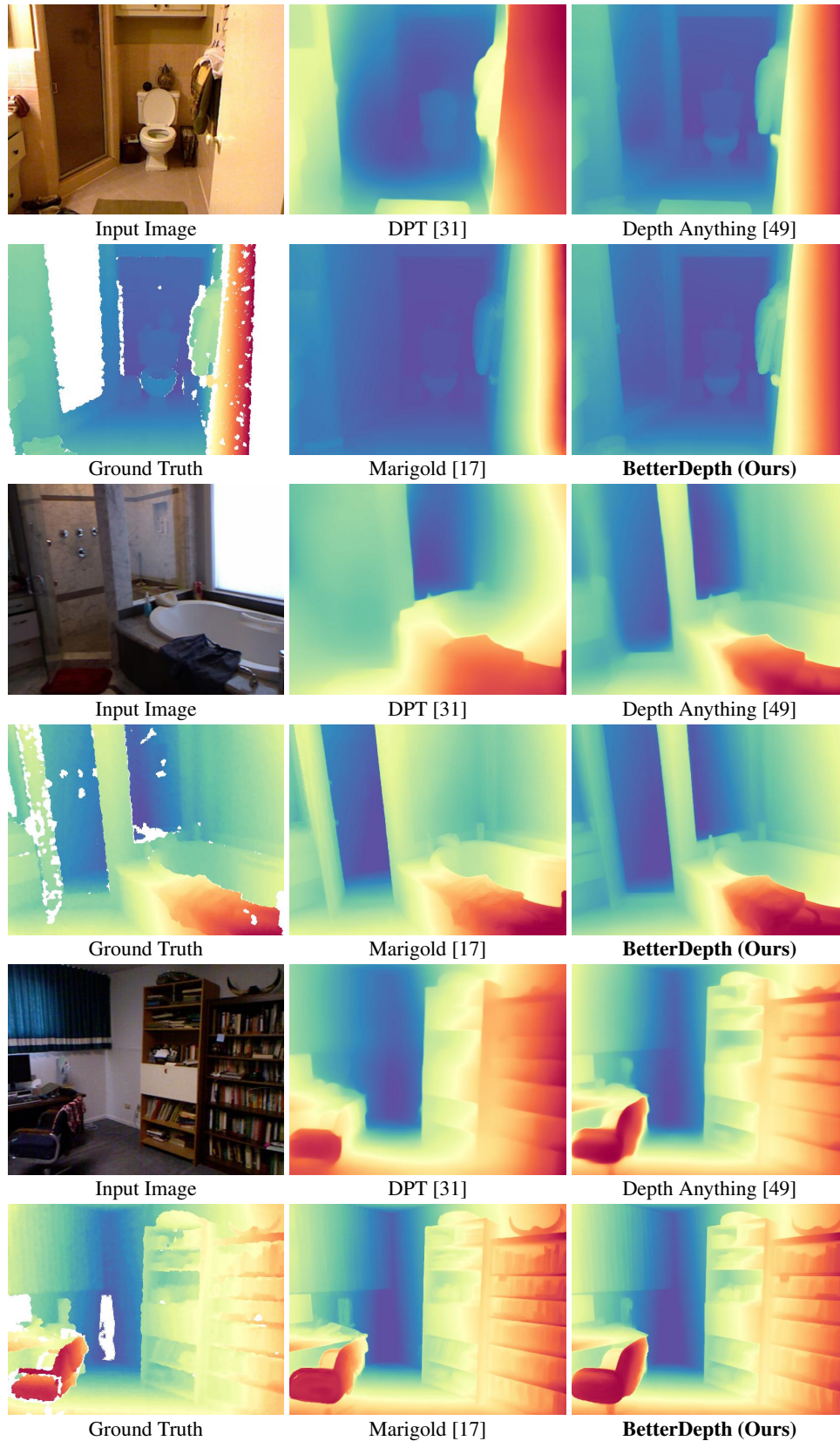| Input Image | DPT [31] | Depth Anything [49] |
Ground Truth	Marigold [17]	**BetterDepth (Ours)**
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**

Figure A9: **Qualitative comparisons on the NYUv2 dataset [28]**, part 2. Predictions are aligned to ground truth. For better visualization, color coding is consistent across all results, where red indicates the close plane and blue means the far plane.
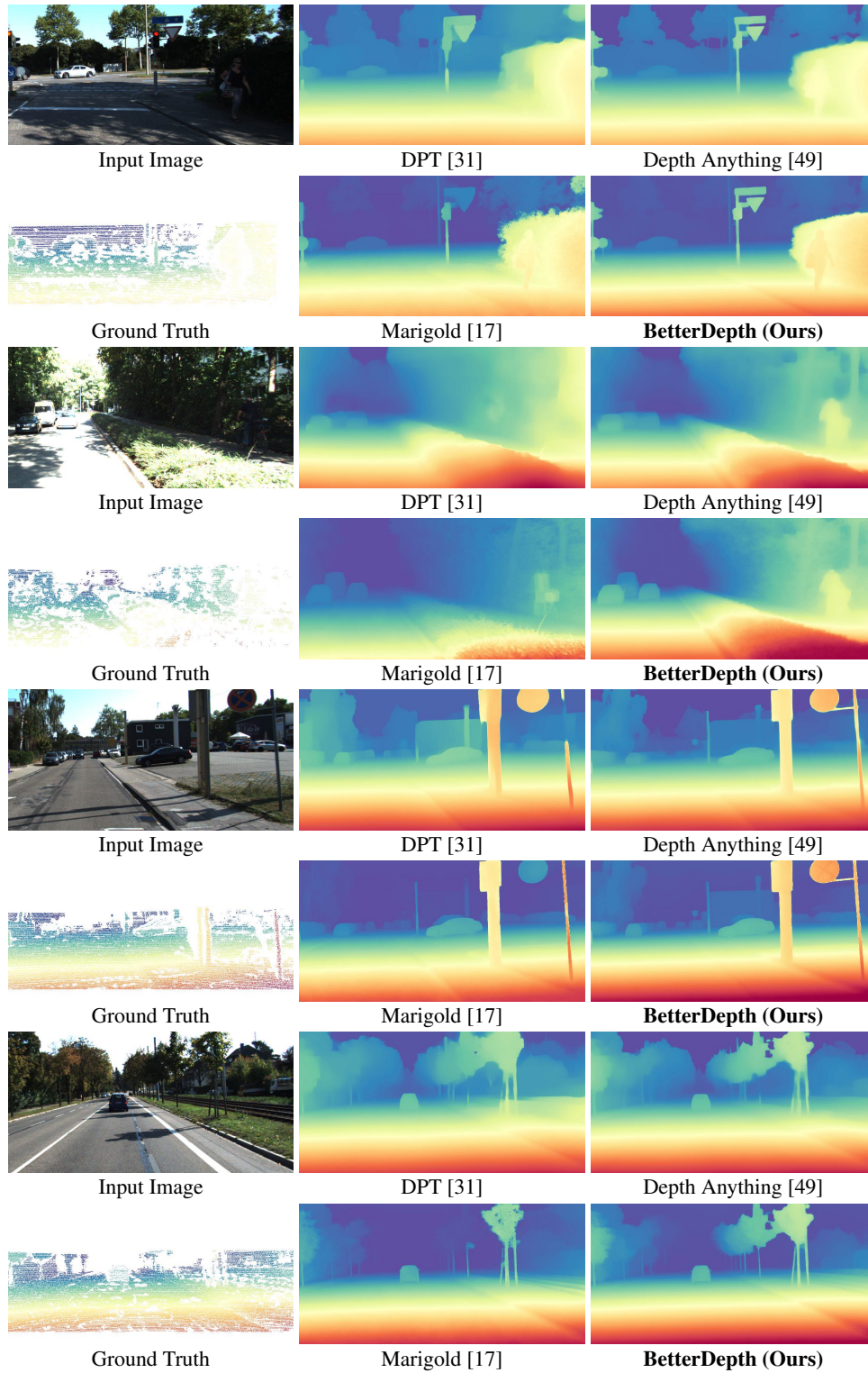
Figure A10: **Qualitative comparisons on the KITTI dataset [11]**, part 1. Predictions are aligned to ground truth. For better visualization, color coding is consistent across all results, where red indicates the close plane and blue means the far plane.
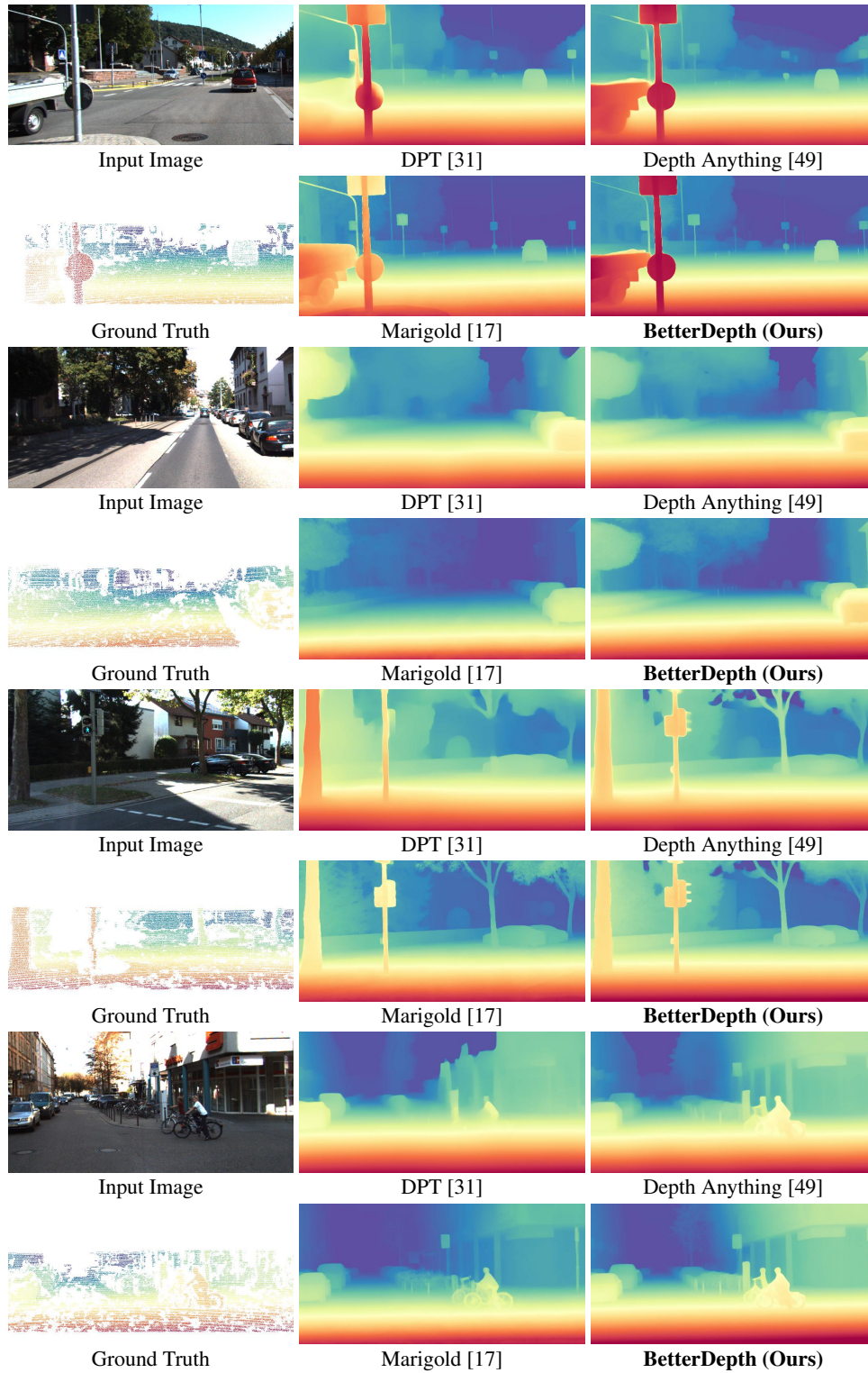
Figure A11: **Qualitative comparisons on the KITTI dataset [11]**, part 2. Predictions are aligned to ground truth. For better visualization, color coding is consistent across all results, where red indicates the close plane and blue means the far plane.

Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**

Figure A12: **Qualitative comparisons on the ETH3D dataset [43]**, part 1. Predictions are aligned to ground truth. For better visualization, color coding is consistent across all results, where red indicates the close plane and blue means the far plane.

| Input Image | DPT [31] | Depth Anything [49] |

| Ground Truth | Marigold [17] | **BetterDepth (Ours)** |

| Input Image | DPT [31] | Depth Anything [49] |

| Ground Truth | Marigold [17] | **BetterDepth (Ours)** |

| Input Image | DPT [31] | Depth Anything [49] |

| Ground Truth | Marigold [17] | **BetterDepth (Ours)** |

Figure A13: **Qualitative comparisons on the ETH3D dataset [43]**, part 2. Predictions are aligned to ground truth. For better visualization, color coding is consistent across all results, where red indicates the close plane and blue means the far plane.
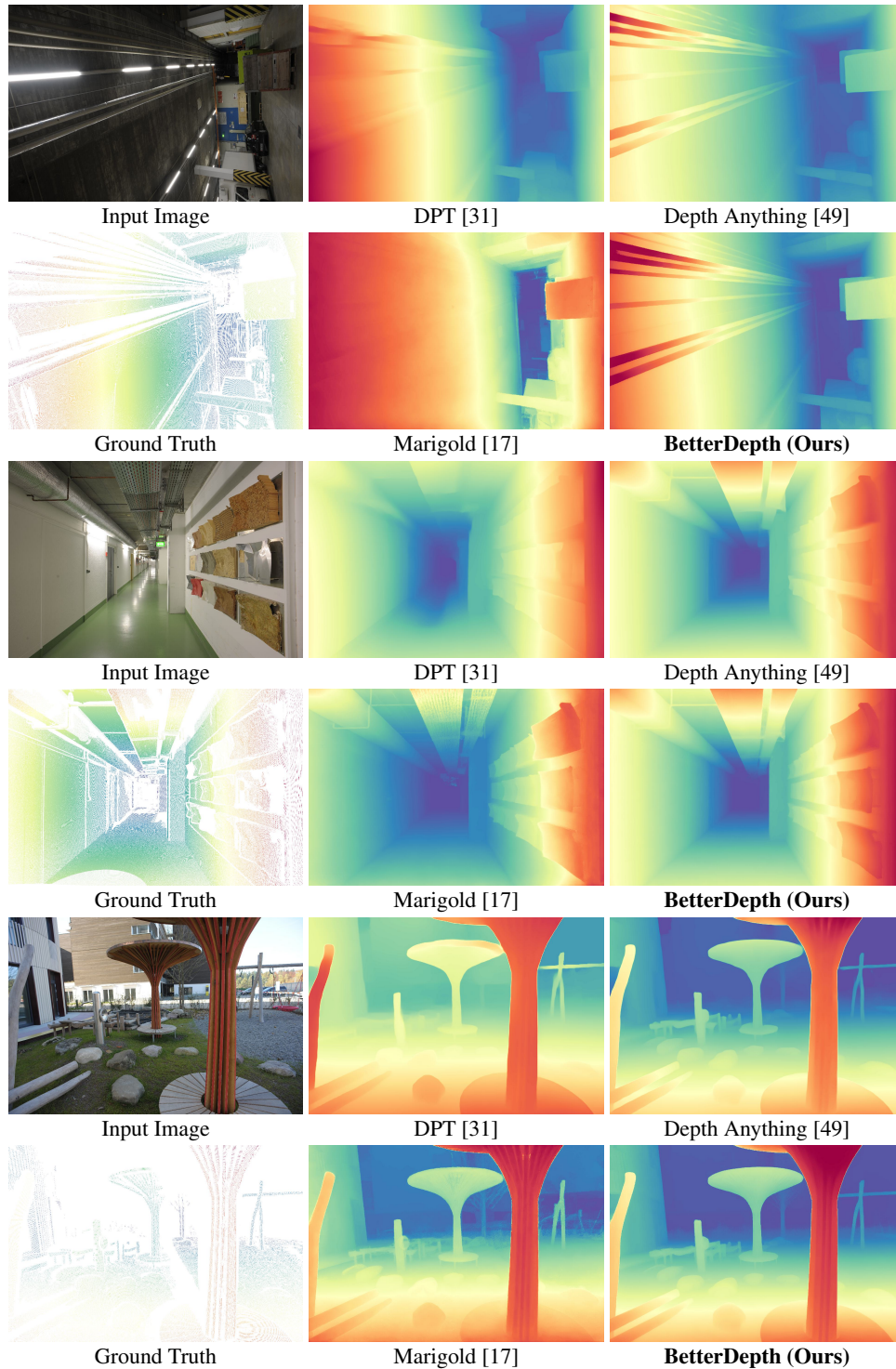
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**

Figure A14: **Qualitative comparisons on the ScanNet dataset [6]**, part 1. Predictions are aligned to ground truth. For better visualization, color coding is consistent across all results, where red indicates the close plane and blue means the far plane.
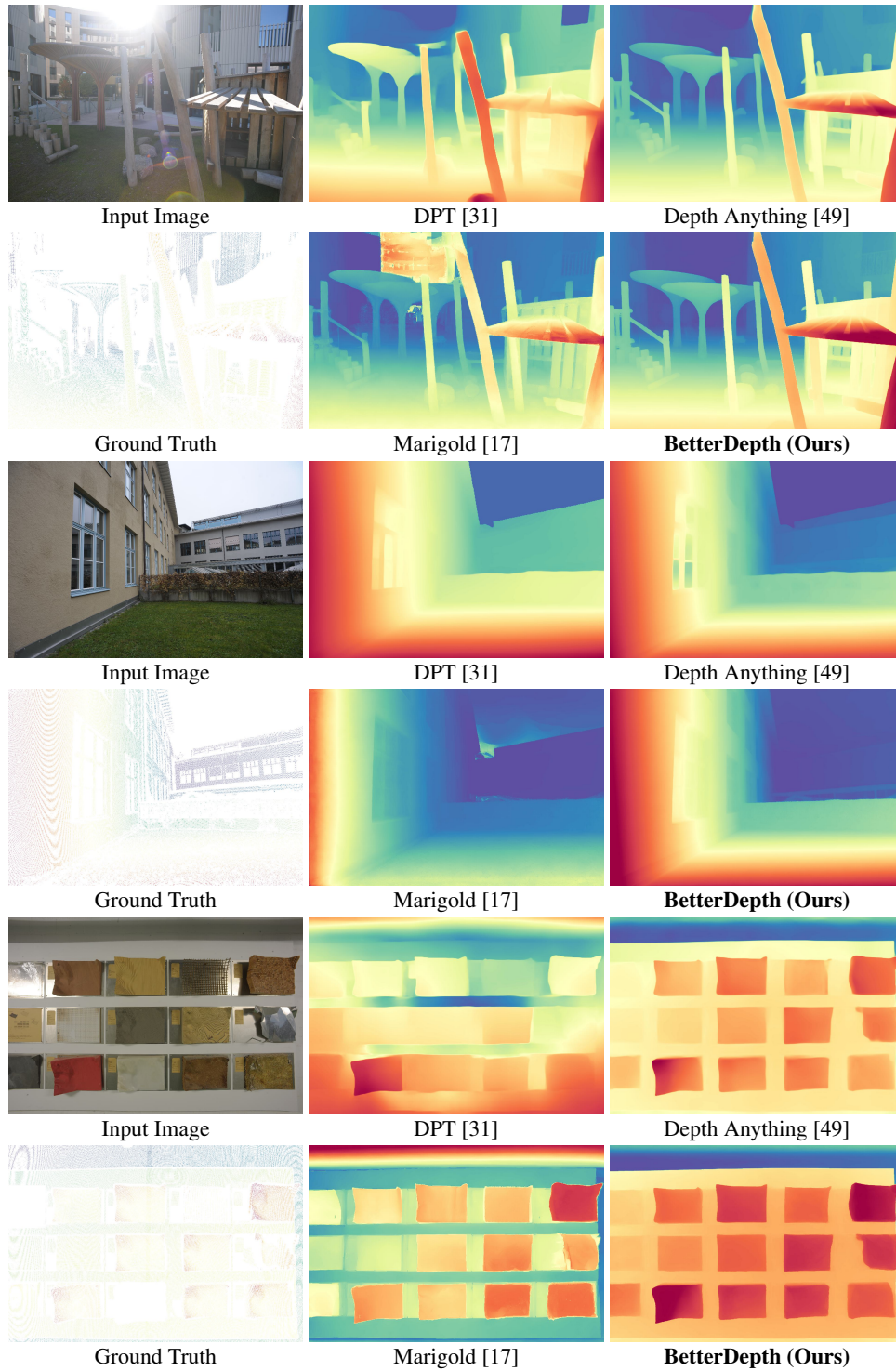
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**

Figure A15: **Qualitative comparisons on the ScanNet dataset [6]**, part 2. Predictions are aligned to ground truth. For better visualization, color coding is consistent across all results, where red indicates the close plane and blue means the far plane.
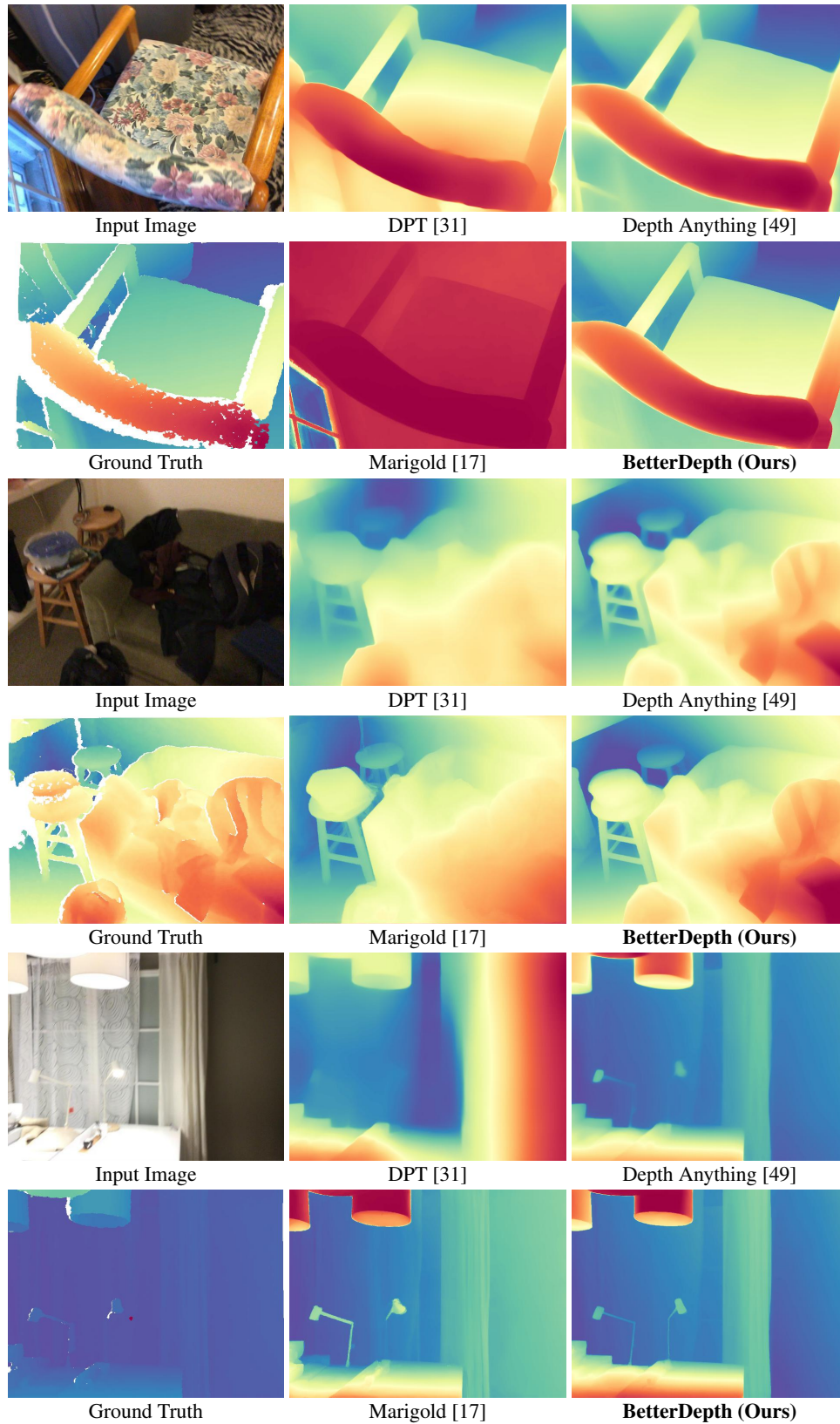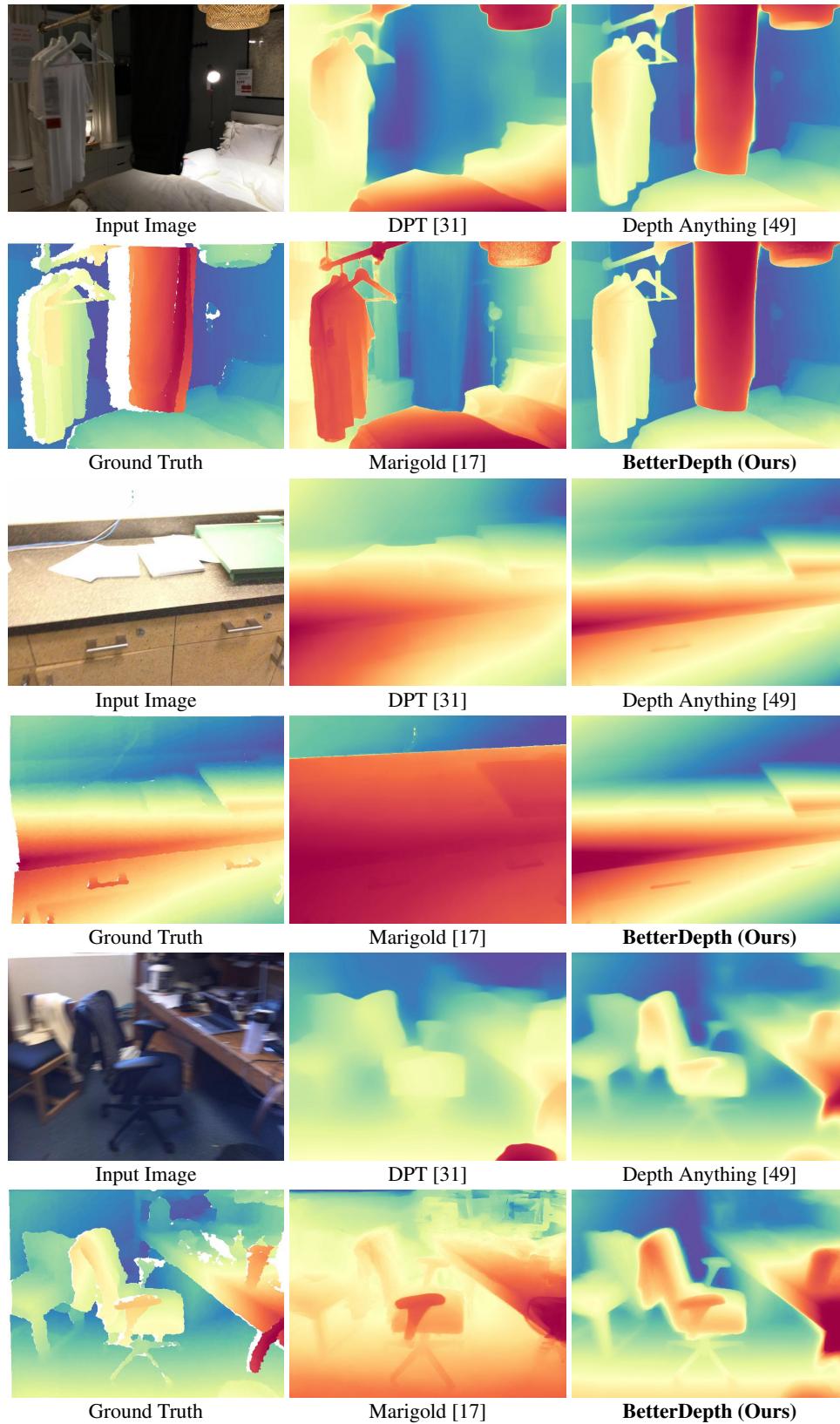
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**

Figure A16: **Qualitative comparisons on the DIODE dataset [45]**, part 1. Predictions are aligned to ground truth. For better visualization, color coding is consistent across all results, where red indicates the close plane and blue means the far plane.
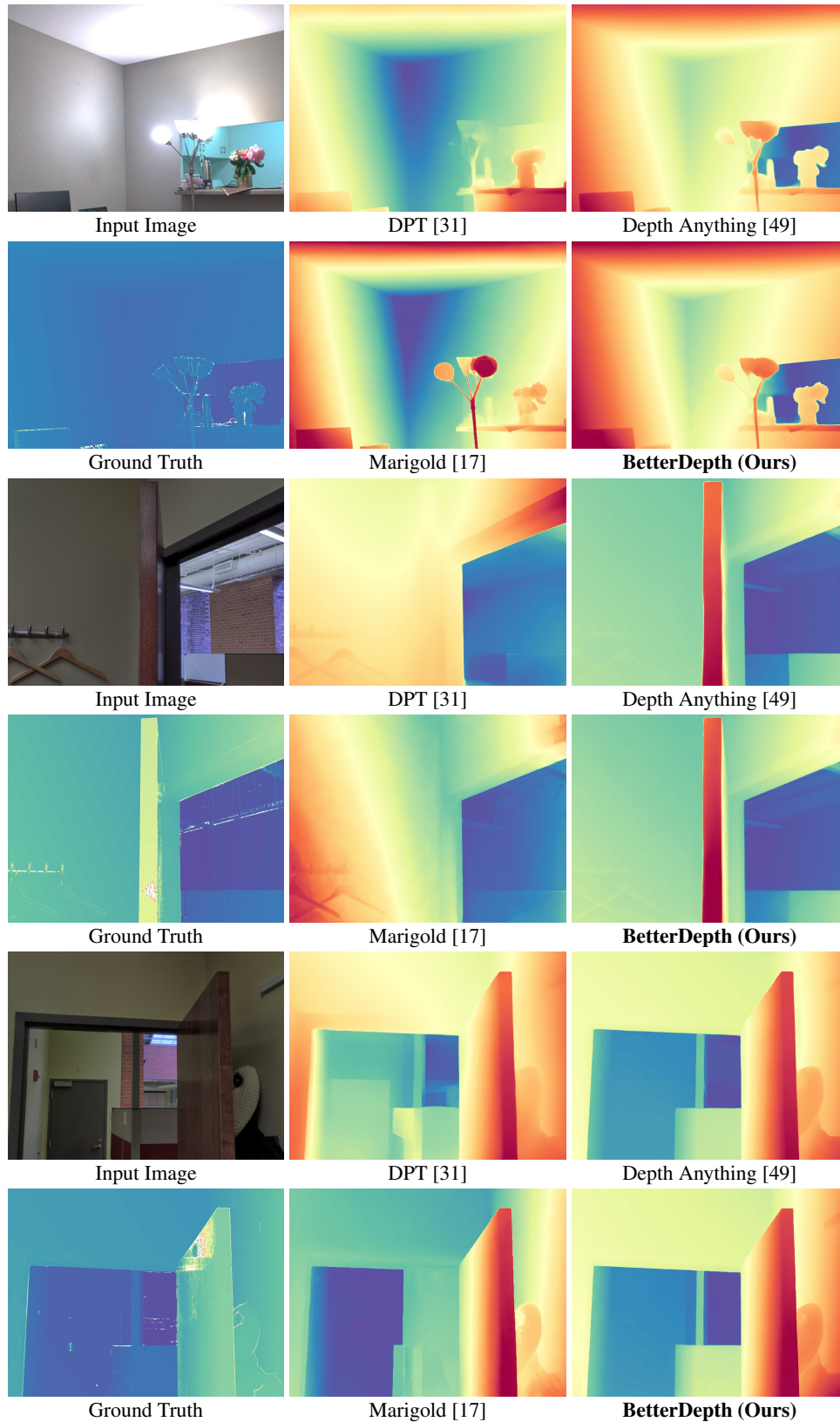
Input Image	DPT [31]	Depth Anything [49]
Ground Truth	Marigold [17]	**BetterDepth (Ours)**

Figure A17: **Qualitative comparisons on the DIODE dataset [45]**, part 2. Predictions are aligned to ground truth. For better visualization, color coding is consistent across all results, where red indicates the close plane and blue means the far plane.
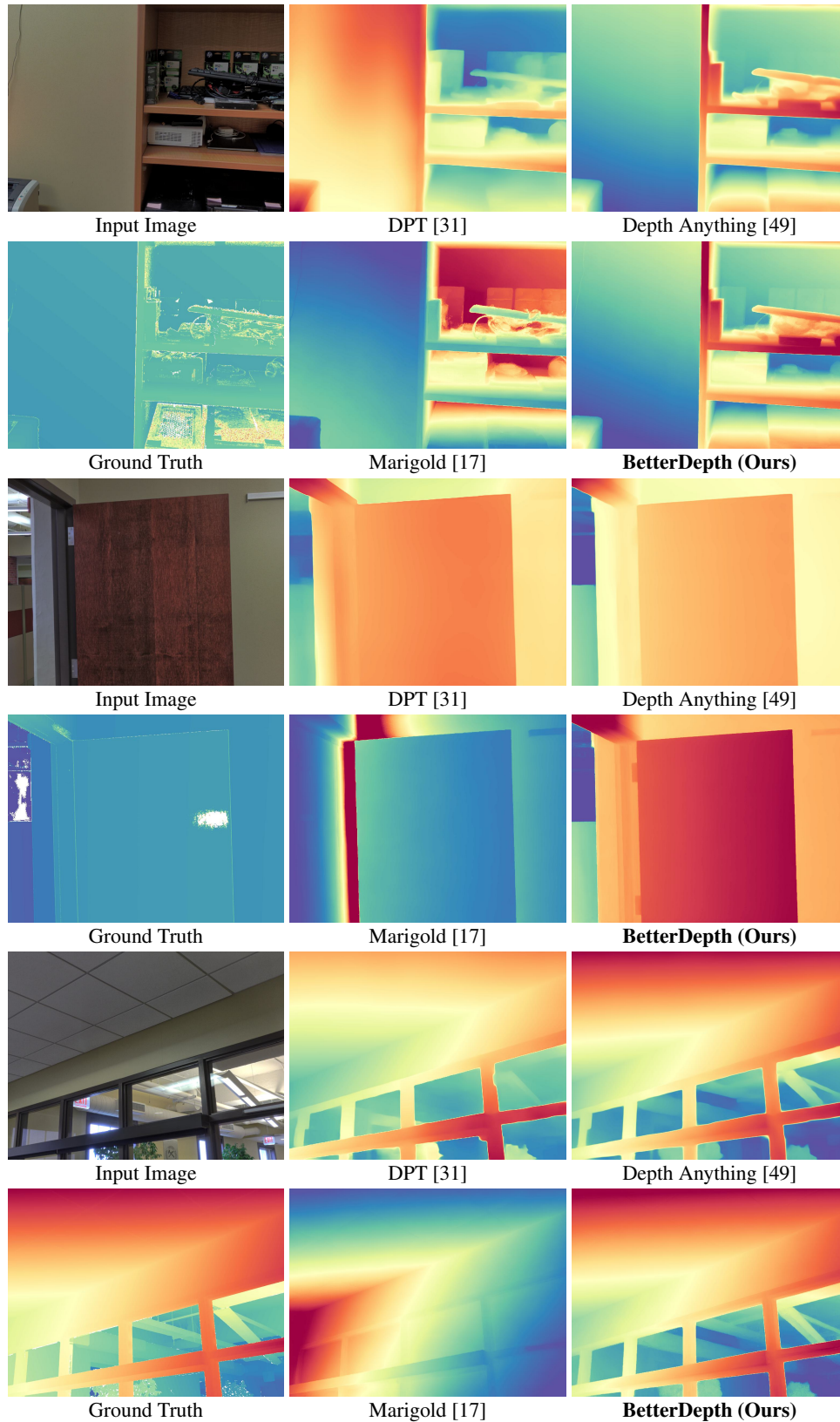
NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect our contribution.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see the Limitation and Future Work section (Sec. I) in the Appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please see the Experiment and Analysis section (Sec. 4).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: This is research done in collaboration with a corporate research lab and we haven't been able to get clearance to release the code. However, implementation is described in detail in the Experiments and Analysis section (Sec. 4) and the data used for training is publicly accessible (Hypersim: `https://github.com/apple/ml-hypersim`, Virtual KITTI: `https://europe.naverlabs.com/research-old2/computer-vision/proxy-virtual-worlds-vkitti-2/`). We additionally provide the detailed training procedure in Algorithm 1 to facilitate reproducing our training models and experimental results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please see the Experiments and Analysis section (Sec. 4) in the main paper and the Hyperparameter Analysis section (Sec. F) in the Appendix.

   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Please see the Error Bar Analysis section (Sec. G) in the Appendix.

   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please see the Experiments and Analysis section (experimental settings in Sec. 4).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Please see the Discussion of Societal Impacts section (Sec. J) in the Appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: This work properly credits and respects the assets used.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Please see the Method (Sec. 3) and the Experiments and Analysis section (Sec. 4).

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.