Pixel-Perfect Depth with Semantics-Prompted Diffusion Transformers

Gangwei Xu[1,2*] **Haotong Lin**[3*] **Hongcheng Luo**[2] **Xianqi Wang**[1] **Jingfeng Yao**[1]
Lianghui Zhu[1] **Yuechuan Pu**[2] **Cheng Chi**[2] **Haiyang Sun**[2†] **Bing Wang**[2]
Guang Chen[2] **Hangjun Ye**[2] **Sida Peng**[3] **Xin Yang**[1†✉]

[1]Huazhong University of Science and Technology [2]Xiaomi EV [3]Zhejiang University
https://pixel-perfect-depth.github.io

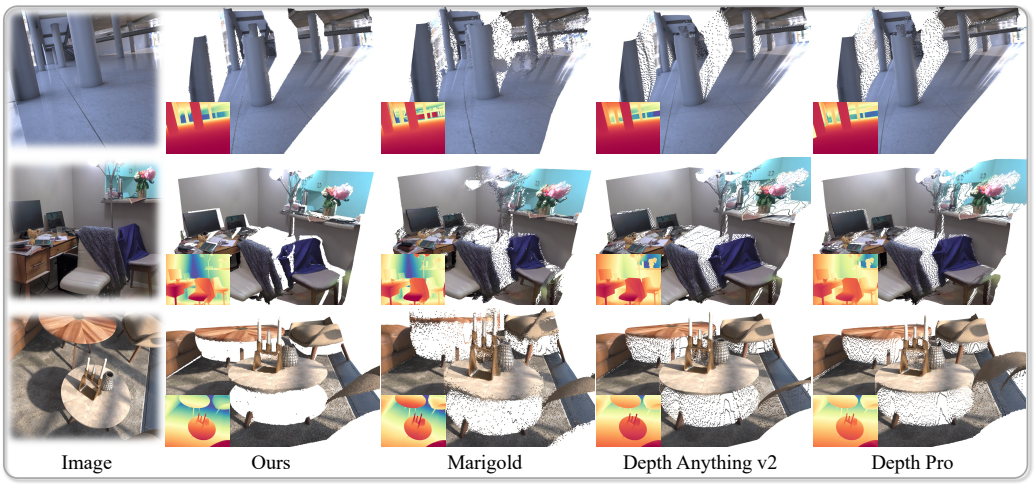

Figure 1: We present **Pixel-Perfect Depth**, a monocular depth estimation model with pixel-space diffusion transformers. Compared to existing discriminative [82, 4] and generative [34] models, its estimated depth maps can produce high-quality, flying-pixel-free point clouds.

Abstract

This paper presents **Pixel-Perfect Depth**, a monocular depth estimation model based on pixel-space diffusion generation that produces high-quality, flying-pixel-free point clouds from estimated depth maps. Current generative depth estimation models fine-tune Stable Diffusion and achieve impressive performance. However, they require a VAE to compress depth maps into the latent space, which inevitably introduces *flying pixels* at edges and details. Our model addresses this challenge by directly performing diffusion generation in the pixel space, avoiding VAE-induced artifacts. To overcome the high complexity associated with pixel-space generation, we introduce two novel designs: 1) **Semantics-Prompted Diffusion Transformers** (**SP-DiT**), which incorporate semantic representations from vision foundation models into DiT to prompt the diffusion process, thereby preserving global semantic consistency while enhancing fine-grained visual details; and 2) **Cascade DiT Design** that progressively increases the number of tokens to further enhance efficiency and accuracy. Our model achieves the best performance among all published generative models across five benchmarks, and significantly outperforms all other models in edge-aware point cloud evaluation.

[*] Equal contribution, [†] Project leader, [✉] Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Figure 2: **Qualitative comparisons**. GT(VAE) denotes the ground truth depth map after VAE reconstruction. Existing generative models [34] use a VAE to compress inputs into the latent space, inevitably introducing *flying pixels* at edges and details. In contrast, our model directly performs diffusion in pixel space, avoiding these issues. Depth maps are visualized on the point clouds.

1 Introduction

Monocular depth estimation (MDE) is a fundamental task with a wide range of downstream applications, such as 3D reconstruction, novel view synthesis, and robotic manipulation. Due to its significance, a large number of depth estimation models [34, 81, 82, 88] have emerged recently. These models achieve high-quality results in most zero-shot scenarios or regions, but suffer from *flying pixels* around object boundaries and fine details when converted into point clouds [39], as shown in Figure 1 and 5, which limits their practical applications in tasks such as free-viewpoint broadcast, robotic manipulation, and immersive content creation.

Current models suffer from the *flying pixels* problem due to different reasons. For discriminative models [82, 4, 88, 30], *flying pixels* mainly arise from their tendency to output an intermediate (*average*) depth value between the foreground and background at depth-discontinuous edges, in order to minimize regression loss. In contrast, generative models [34, 17, 23] bypass direct regression by modeling pixel-wise depth distributions, allowing them to preserve sharp edges and recover fine structures more faithfully. However, current generative depth models typically fine-tune Stable Diffusion [51] for depth estimation, which requires a Variational Autoencoder (VAE) to compress depth maps into a latent space. This compression inevitably leads to the loss of edge sharpness and structural fidelity, resulting in a significant number of *flying pixels*, as shown in Figure 2.

A trivial solution could be training a diffusion-based monocular depth model in pixel space, bypassing the use of a VAE. However, we find this highly challenging, due to the increased complexity and instability of modeling both global semantic consistency and fine-grained visual details, leading to extremely low-quality depth predictions (Table 2 and Figure 6). To further investigate this limitation, we examine prior studies on high-resolution image generation. Several works [29, 61, 94], through signal-to-noise ratio (SNR) analysis, have pointed out that adding noise with higher intensity is more likely to disrupt the global structures or low-frequency components of high-resolution images, thereby improving generation. This reveals that the primary difficulty in high-resolution pixel-space generation lies in effectively perceiving and modeling global image structures.

In this paper, we present **Pixel-Perfect Depth**, a framework for high-quality and flying-pixel-free monocular depth estimation using pixel-space diffusion transformers. Recognizing that the major difficulty in high-resolution pixel-space generation lies in perceiving and modeling global image structures. To address this challenge, we propose the **Semantics-Prompted Diffusion Transformers** (**SP-DiT**) that incorporate high-level semantic representations into the diffusion process to enhance the model's ability to preserve global structures and semantic coherence. Equipped with SP-DiT, our model can more effectively preserve global semantic consistency while generating fine-grained visual details in high-resolution pixel space. However, the semantic representations obtained from vision foundation models [44, 82, 65, 24] often do not align well with the internal representations of DiT, leading to training instability and convergence issues. To address this, we introduce a simple yet effective regularization technique for semantic representations, which ensures stable training and facilitates convergence to desirable solutions. As shown in Table 2 and Figure 6, SP-DiT significantly improves overall performance, with up to a 78% gain on the NYUv2 [58] AbsRel metric.

Furthermore, we introduce the **Cascade DiT Design** (Cas-DiT), an efficient architecture for diffusion transformers. We find that in diffusion transformers, the early blocks are primarily responsible for capturing and generating global or low-frequency structures, while the later blocks focus on

generating high-frequency details. Based on this insight, Cas-DiT adopts a progressive patch size strategy: larger patch size is used in the early DiT blocks to reduce the number of tokens and facilitate global image structure modeling; in the later DiT blocks, we increase the number of tokens, which is equivalent to using a smaller patch size, allowing the model to focus on the generation of fine-grained spatial details. This coarse-to-fine cascaded design not only significantly reduces computational costs and improves efficiency, but also delivers substantial improvements in accuracy.

We highlight the main contributions of this paper below:

- We present **Pixel-Perfect Depth**, a monocular depth estimation model with pixel-space diffusion generation, capable of producing flying-pixel-free point clouds from estimated depth maps.
- We introduce **Semantics-Prompted DiT**, which integrates normalized semantic representations into the DiT to effectively preserve global semantic consistency while enhancing fine-grained visual details. This significantly boosts overall performance. We further propose a novel **Cascade DiT Design** to enhance the efficiency and accuracy of our model.
- Our model achieves the best performance across five benchmarks among all published generative depth estimation models.
- We introduce an edge-aware point cloud evaluation metric, which effectively assesses *flying pixels* at edges. Our model significantly outperforms previous models in this evaluation.

2 Related Work

2.1 Monocular Depth Estimation

Depth estimation can be broadly categorized into monocular [82, 69], stereo [75, 72, 77, 76, 22, 8, 7, 9], and sparse depth completion [41] methods. Early monocular depth estimation methods relied primarily on manually designed features [52, 28]. The advent of neural networks revolutionized the field, though initial approaches [15, 14] struggled with cross-dataset generalization. To address this limitation, scale-invariant and relative loss [49] are introduced, enabling multi-dataset [36, 86, 10, 74, 71, 68, 64, 73, 50, 40] training. Recent methods focus on improving the generalization ability [82, 4, 66, 67], depth consistency [80, 6, 31, 33], and metric scale [3, 37, 38, 88, 21, 89, 30, 46, 41] of depth estimation. These methods converge towards using transformer-based architectures [48]. Concurrent works [69, 70, 79] explore point cloud representations to improve depth estimation performance. Several recent methods [32, 12, 55, 53, 54, 93] have attempted to use diffusion models for metric depth estimation. In contrast, our method focuses on relative depth and demonstrates improved generalization and fine-grained detail across a wide range of real-world scenes. Furthermore, our model significantly differs from these methods by introducing Semantics-Prompted DiT, which incorporates pretrained high-level semantic representations into the diffusion process, greatly enhancing performance.

More recently, [34] brought the new insight to the field by fine-tuning pretrained Stable Diffusion [51] for depth estimation, which demonstrated impressive zero-shot capabilities for relative depth. The following works [23, 20, 60, 92, 2] attempt to improve its performance and inference speed. However, they are all based on the latent diffusion model [51], which is trained in the latent space and requires a VAE to compress the depth map into a latent space. We focus on a pixel-space diffusion model that is trained directly in the pixel space without requiring any VAE.

2.2 Diffusion Generative Models

Diffusion generative models [25, 59, 45, 90, 84, 85, 96] have demonstrated impressive results in image and video generation. Early approaches [25, 27, 26] such as DDPM [25] operate directly in the pixel space, enabling high-fidelity generation but incurring significant computational costs, especially at high resolutions. To address this limitation, Latent Diffusion Models perform the diffusion process in a lower-dimensional latent space obtained via a VAE, as popularized by Stable Diffusion [51]. This design significantly improves training and inference efficiency and has been widely adopted in recent works [16, 85, 90, 95, 35, 47, 83].

Diffusion models for monocular depth estimation typically follow a similar trend. For instance, Marigold [34] and its follow-ups [23, 20] fine-tune pretrained Stable Diffusion [51] models for depth

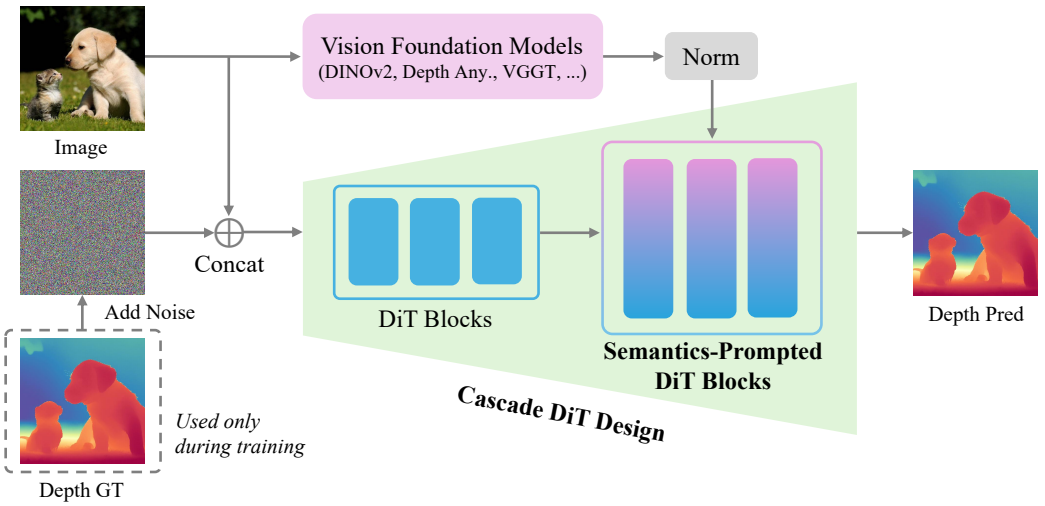

Figure 3: **Overview of Pixel-Perfect Depth.** Given an input image, we concatenate it with noise and feed it into the proposed Cascade DiT. Meanwhile, the image is also processed by a pretrained encoder from Vision Foundation Models to extract high-level semantics, forming our Semantics-Prompted DiT. We perform diffusion generation directly in pixel space without using any VAE.

estimation, benefiting from fast convergence and strong priors learned from large-scale datasets. However, the VAE's latent compression leads to *flying pixels* in the resulting point clouds. In contrast, pixel-space diffusion avoids such artifacts but remains computationally intensive and slow to converge at high resolutions. To address this, we propose Semantics-Prompted DiT and Cascade DiT Design, which enables efficient high-resolution depth estimation without latent compression.

3 Method

3.1 Pixel-Perfect Depth

Given an input image, our goal is to estimate a pixel-perfect depth map that is free of *flying pixels* when converted to point clouds. Existing models [34, 17, 23, 82, 4] often suffer from *flying pixels* due to their inherent modeling paradigms. Discriminative models tend to smooth object edges and blur fine details because of their mean-prediction bias, which results in noticeable *flying pixels* in the reconstructed point clouds. Generative models, in theory, can better capture the multi-modal depth distribution at object edges. However, current generative models typically fine-tune Stable Diffusion [51] for depth estimation, relying on its strong image priors. This requires compressing the depth map into a latent space via a VAE, inevitably causing *flying pixels*.

To unleash the potential of generative models for depth estimation, we propose **Pixel-Perfect Depth** that performs diffusion directly in the pixel space instead of the latent space. It allows us to directly model the pixel-wise distribution of depth, such as the discontinuities at object edges. However, training a generative diffusion model directly in the high-resolution pixel space (*e.g.*, 1024×768) is computationally demanding and hard to optimize. To overcome these challenges, we introduce Semantics-Prompted DiT and Cascaded DiT Design, detailed in the following sections.

3.2 Generative Formulation

We adopt Flow Matching [42, 43, 1] as the generative core of our depth estimation framework. Flow Matching learns a continuous transformation from Gaussian noise to a data sample via a first-order Ordinary Differential Equation (ODE). In our case, we model the transformation from Gaussian noise to a depth sample. Specifically, given a clean depth sample $\mathbf{x}_0 \sim \mathcal{D}$ and Gaussian noise $\mathbf{x}_1 \sim \mathcal{N}(0, 1)$, we define an interpolated sample at continuous time $t \in [0, 1]$ as:

$$\mathbf{x}_t = t \cdot \mathbf{x}_1 + (1 - t) \cdot \mathbf{x}_0. \tag{1}$$

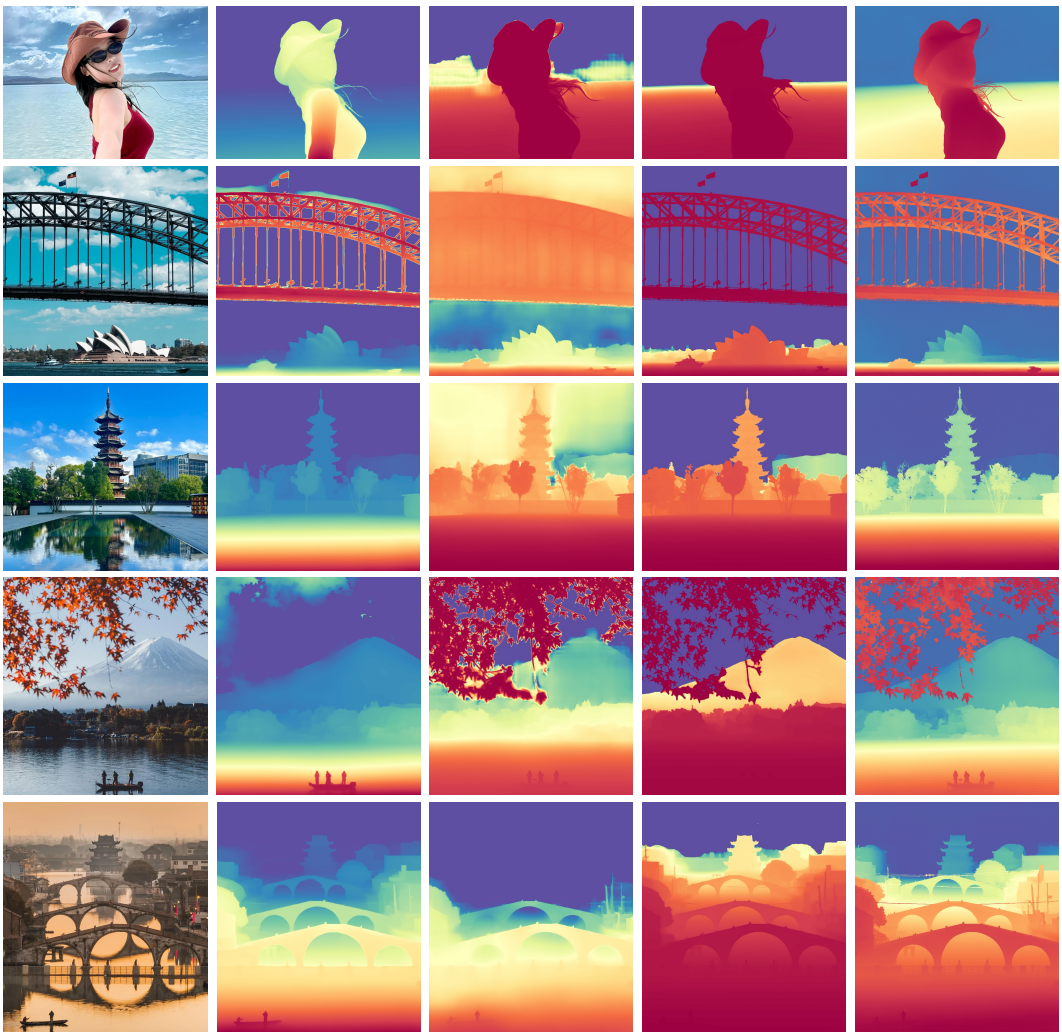

Figure 4: **Comparison with existing depth foundation models on open-world images.** Our model preserves more fine-grained details than Depth Anything v2 [82] and MoGe 2 [70], while demonstrating significantly higher robustness compared to Depth Pro [4].

This defines a velocity field:

$$\mathbf{v}_t = \frac{d\mathbf{x}_t}{dt} = \mathbf{x}_1 - \mathbf{x}_0, \tag{2}$$

which describes the direction from clean data to noise. Our model $\mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{c})$ is trained to predict the velocity field, based on the current noisy sample $\mathbf{x}_t$, the time step t, and the input image $\mathbf{c}$. The training objective is the mean squared error (MSE) between the predicted and true velocity:

$$\mathcal{L}_{\text{velocity}(\theta)} = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1, t}\left[\left\|\mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{c}) - \mathbf{v}_t\right\|^2\right]. \tag{3}$$

At inference, we start from noise $\mathbf{x}_1$ and solve the ODE by discretizing the time interval $[0, 1]$ into steps t_i, iteratively updating the depth sample as follows:

$$\mathbf{x}_{t_{i-1}} = \mathbf{x}_{t_i} + \mathbf{v}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})(t_{i-1} - t_i), \tag{4}$$

where t_i decreases from 1 to 0, gradually transforming the initial noise $\mathbf{x}_1$ into the depth sample $\mathbf{x}_0$.

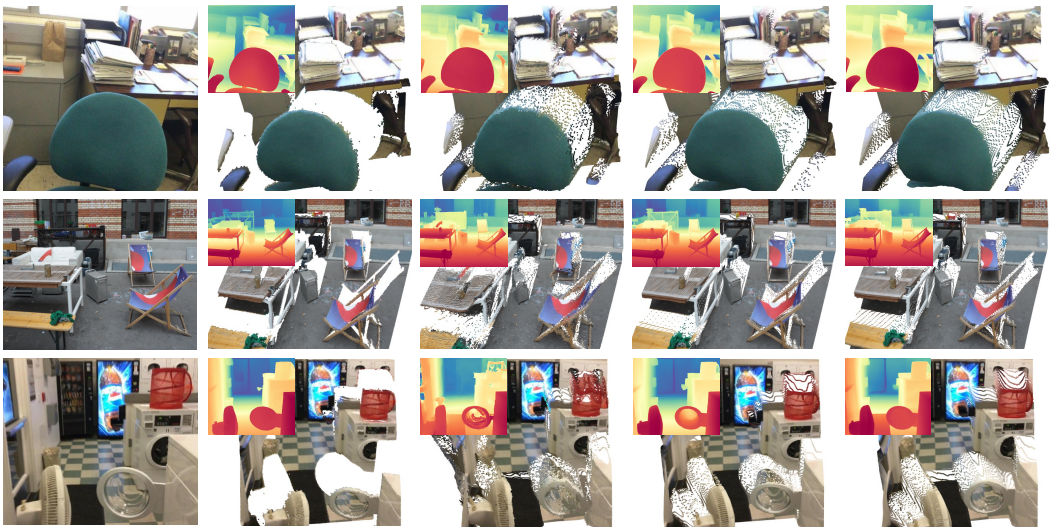

Figure 5: **Qualitative point cloud results in complex scenes.** Our model produces significantly fewer *flying pixels* compared to other depth estimation models [34, 82, 4], with depth maps overlaid on the point clouds for visualization.

3.3 Semantics-Prompted Diffusion Transformers

Our Semantics-Prompted DiT builds on DiT [45] for its simplicity, scalability, and strong performance in generative modeling. Unlike previous depth estimation models such as Depth Anything v2 [82] and Marigold [34], our architecture is purely transformer-based, without any convolutional layers. By integrating high-level semantic representations, SP-DiT enables our model to preserve global semantic consistency while enhancing fine-grained visual details, without sacrificing the simplicity and scalability of DiT.

Specifically, given the interpolated noise sample $\mathbf{x}_t$ and the corresponding image $\mathbf{c}$, we first concatenate them into a single input: $\mathbf{a}_t = \mathbf{x}_t \oplus \mathbf{c}$, where the image $\mathbf{c}$ serves as a condition. Then, we directly feed $\mathbf{a}_t$ into the DiT. The first layer of DiT is a patchify operation, which converts the spatial input $\mathbf{a}_t$ into a 1D sequence of T tokens (patches), each with a dimension of D, by linearly embedding each patch of size $p \times p$ from the input $\mathbf{a}_t$. Subsequently, the input tokens are processed by a sequence of Transformer blocks, called DiT blocks. After the final DiT block, each token is linearly projected into a $p \times p$ tensor, which is then reshaped back to the original spatial resolution to obtain the predicted velocity $\mathbf{v}_t$ (*i.e.*, $\mathbf{x}_1 - \mathbf{x}_0$), with a channel dimension of 1.

Unfortunately, performing diffusion directly in the pixel space leads to poor convergence and highly inaccurate depth predictions. As shown in Figure 6, the model struggles to model both global image structure and fine-grained details. To address this, we extract high-level semantic representations $\mathbf{e}$ as guidance from the input image $\mathbf{c}$ using a vision foundation model f, as follows:

$$\mathbf{e} = f(\mathbf{c}) \in \mathbb{R}^{T' \times D'}, \tag{5}$$

where T' and D' are the number of tokens and the embedding dimension of $f(\mathbf{c})$, respectively. These high-level semantic representations are then incorporated into our DiT model, enabling it to more effectively preserve global semantic consistency while enhancing fine-grained visual details. However, we found that the magnitude of the obtained semantics $\mathbf{e}$ differs significantly from the magnitude of the tokens in our DiT model, which affects both the stability of the model's training and its performance. To address this, we normalize the semantic representation $\mathbf{e}$ along the feature dimension using L2 norm, as follows:

$$\hat{\mathbf{e}} = \frac{\mathbf{e}}{\|\mathbf{e}\|_2}. \tag{6}$$

Subsequently, the normalized semantic representation is integrated into the tokens $\mathbf{z}$ of our DiT model via a multilayer perceptron (MLP) layer h_ϕ,

$$\mathbf{z}' = h_\phi(\mathbf{z} \oplus \mathcal{B}(\hat{\mathbf{e}})), \tag{7}$$

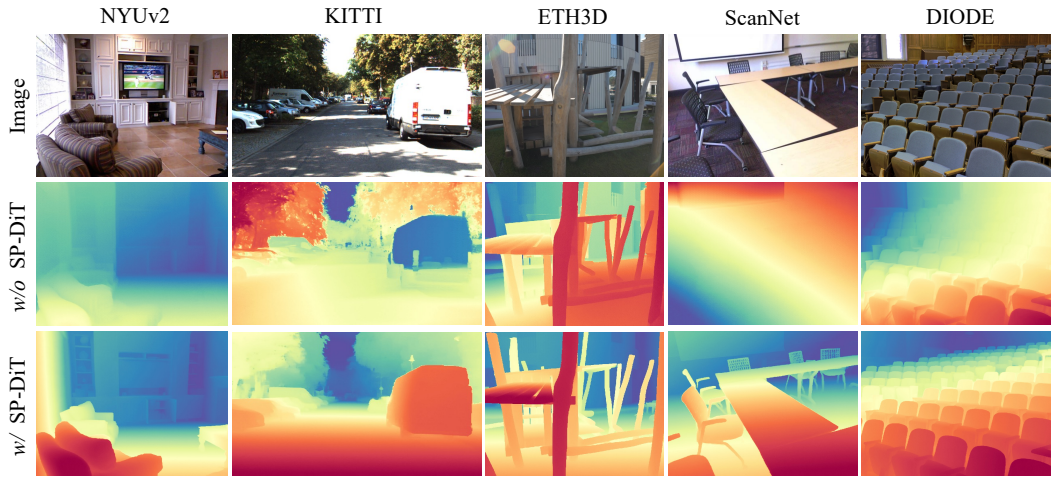

Figure 6: **Qualitative ablations for the proposed SP-DiT.** Without SP-DiT, the vanilla DiT model struggles with preserving global semantics and generating fine-grained visual details.

where $\mathcal{B}(\cdot)$ denotes the bilinear interpolation operator, which aligns the spatial resolution of the semantic representation $\hat{\mathbf{e}}$ with that of the DiT tokens. The resulting $\mathbf{z}'$ denotes the DiT tokens enhanced with semantics. After the fusion, the subsequent DiT blocks are prompted by semantics to effectively preserve global semantic consistency while enhancing fine-grained visual details in the high-resolution pixel space. We refer to these subsequent DiT blocks as Semantics-Prompted DiT.

In this work, we experiment with various pretrained vision foundation models, including DINOv2 [44], VGGT [65], MAE [24], and Depth Anything v2 [82]. All of them significantly boost performance and facilitate more stable and efficient training, as shown in Table 3. Note that we only utilize the encoder of each vision foundation model, *e.g.*, a 24-layer Vision Transformer encoder (ViT-L/14) for both DINOv2 [44] and Depth Anything v2 [82].

3.4 Cascade DiT Design

Although the proposed Semantics-Prompted DiT significantly improves accuracy performance, performing diffusion directly in the pixel space remains computationally expensive. To address this issue, We propose a novel Cascaded DiT Design to reduce the computational burden of the model. We observe that in DiT architectures, the early blocks are primarily responsible for capturing global image structures and low-frequency information, while the later blocks focus on modeling fine-grained, high-frequency details.

To optimize the efficiency and effectiveness of this process, we adopt a large patch size in the early DiT blocks. This design significantly reduces the number of tokens that need to be processed, leading to lower computational cost. Additionally, it encourages the model to prioritize learning and modeling global image structures and low-frequency information, which also better aligns with the high-level semantic representations extracted from the input image. In the later DiT blocks, we increase the number of tokens, which is equivalent to using a smaller patch size. This allows the model to better focus on fine-grained spatial details. The resulting coarse-to-fine cascaded design mirrors the hierarchical nature of visual perception and improves both the efficiency and accuracy of depth estimation.

Specifically, for our diffusion model with a total of N DiT blocks, the first $N/2$ blocks constitute the coarse stage with a larger patch size, while the remaining $N/2$ blocks (*i.e.*, SP-DiT) form the fine stage using a smaller patch size.

3.5 Implementation Details

In this section, we provide essential information about the model architecture details, depth normalization, and training details.

Table 1: **Zero-shot relative depth estimation.** Better: AbsRel ↓, δ_1 ↑. **Bold** numbers are the best. Our model outperforms other generative models on five benchmarks. Ours (512) represents 512×512 model, and Ours (1024) represents 1024×768 model.

Type	Method	Training Data	NYUv2		KITTI		ETH3D		ScanNet		DIODE	
			AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑
Discriminative	DiverseDepth[87]	320K	11.7	87.5	19.0	70.4	22.8	69.4	10.9	88.2	-	-
	MiDaS[49]	2M	11.1	88.5	23.6	63.0	18.4	75.2	12.1	84.6	-	-
	LeReS[89]	354K	9.0	91.6	14.9	78.4	17.1	77.7	9.1	91.7	-	-
	Omnidata[13]	12M	7.4	94.5	14.9	83.5	16.6	77.8	7.5	93.6	-	-
	HDN[91]	300K	6.9	94.8	11.5	86.7	12.1	83.3	8.0	93.9	-	-
	DPT[48]	1.2M	9.8	90.3	10.0	90.1	**7.8**	**94.6**	8.2	93.4	-	-
	DepthAny. v2[82]	54K	5.4	97.2	8.6	92.8	12.3	88.4	-	-	8.8	93.7
	DepthAny. v2[82]	62M	**4.5**	**97.9**	**7.4**	**94.6**	13.1	86.5	**6.5**	**97.2**	**6.6**	**95.2**
Generative	Marigold[34]	74K	5.5	96.4	9.9	91.6	6.5	96.0	6.4	95.1	10.0	90.7
	GeoWizard[17]	280K	5.2	96.6	9.7	92.1	6.4	96.1	6.1	95.3	12.0	89.8
	DepthFM[20]	74K	5.5	96.3	8.9	91.3	5.8	96.2	6.3	95.4	-	-
	GenPercept[78]	90K	5.2	96.6	9.4	92.3	6.6	95.7	5.6	96.5	-	-
	Lotus[23]	54K	5.4	96.8	8.5	92.2	5.9	97.0	5.9	95.7	9.8	92.4
	Ours (512)	54K	4.3	97.4	8.0	93.1	4.5	97.7	**4.5**	**97.3**	7.0	95.5
	Ours (1024)	125K	**4.1**	**97.7**	**7.0**	**95.5**	**4.3**	**98.0**	4.6	97.2	**6.8**	**95.9**

Model architecture details. In our implementation, we use a total of $N = 24$ DiT blocks, each operating at a hidden dimension of $D = 1024$. The first 12 blocks are standard DiT blocks with a patch size of 16, corresponding to $(H/16) \times (W/16)$ tokens for an input of size $H \times W$. After the 12th block, we employ an MLP layer to expand the hidden dimension by a factor of 4, followed by reshaping to obtain $(H/8) \times (W/8)$ tokens. The remaining 12 SP-DiT blocks then further process these $(H/8) \times (W/8)$ tokens. Finally, we employ an MLP layer followed by a reshaping operation to transform the processed tokens into an $H \times W$ depth map. In contrast to prior monocular depth models, such as Depth Anything and Depth Pro, our model does not rely on any convolutional layers.

Depth normalization. The ground truth depth values are normalized to match the scale expected by the diffusion model. Before normalization, we convert the depth values into log scale to ensure a more balanced capacity allocation across both indoor and outdoor scenes. Specifically, we apply the transformation $\tilde{\mathbf{d}} = \log(\mathbf{d} + \epsilon)$, where $\tilde{\mathbf{d}}$ denotes the transformed depth, $\mathbf{d}$ is the original depth value, and ϵ is a small positive constant (*e.g.*, 1) to ensure numerical stability. We then normalize the log-scaled depth $\tilde{\mathbf{d}}$ using:

$$\hat{\mathbf{d}} = \frac{\tilde{\mathbf{d}} - d_{\min}}{d_{\max} - d_{\min}} - 0.5, \tag{8}$$

where d_{min} and d_{max} are the 2% and 98% depth percentiles of each map, respectively.

Training details. We train two variants of the diffusion model at different resolutions: one at 512×512 and the other at 1024×768. We train all models on 8 NVIDIA GPUs with a per-GPU batch size of 4, using the AdamW optimizer with a constant learning rate of 1×10^{-4}. The training loss is the MSE loss between the predicted and true velocity, as shown in Equation 3, and the gradient matching loss, which is adopted from [82].

4 Experiments

4.1 Experimental Setup

Training datasets. Our objective is to estimate pixel-perfect depth maps, which, when converted to point clouds, are free of *flying pixels* and geometric artifacts. To achieve this, it is essential to train on datasets with high-quality ground truth point clouds. We adopt Hypersim [50], a photorealistic synthetic dataset with accurate and clean 3D geometry, which contains approximately 54K samples, to train the 512×512 model. For the 1024×768 model, we additionally leverage four datasets, UrbanSyn [19] (7.5K), UnrealStereo4K [62] (8K), VKITTI [5] (25K), and TartanAir [71] (30K), to further enhance the model's generalization and robustness.

Table 2: **Ablation studies on five zero-shot benchmarks.** All metrics are presented in percentage terms, **bold** numbers are the best. Inference time was tested on an RTX 4090 GPU. All results were obtained using the 512×512 model.

Method	NYUv2		KITTI		ETH3D		ScanNet		DIODE		Time(s)
	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	
DiT (vanilla)	22.5	72.8	27.3	63.9	12.1	87.4	25.7	65.1	23.9	76.5	0.19
SP-DiT	4.8	96.7	8.6	92.2	4.6	97.5	6.2	94.8	8.2	94.1	0.20
SP-DiT+Cas-DiT	**4.3**	**97.4**	**8.0**	**93.1**	**4.5**	**97.7**	**4.5**	**97.3**	**7.0**	**95.5**	**0.14**

Evaluation setup. Following the majority of previous depth estimation models [34, 17, 23], we evaluate the zero-shot relative depth estimation performance on five real-world datasets: NYUv2 [58], KITTI [18], ETH3D [56], ScanNet [11], and DIODE [63], covering both indoor and outdoor scenes. To assess the quality of depth estimation, we adopt two widely-used evaluation metrics: Absolute Relative Error (AbsRel) and δ_1 accuracy. To demonstrate that our model generates point clouds without *flying pixels*, we convert the estimated depth maps into 3D point clouds and evaluate them using the proposed edge-aware metric. For simplicity, the majority of quantitative evaluations are conducted using the 512×512 model. We employ the 1024×768 model for the quantitative evaluations in Table 1 as well as for qualitative comparisons.

4.2 Ablations and Analysis

Component-wise ablation analysis. We adopt the vanilla DiT [45] model as our baseline and conduct ablations on our proposed modules. Quantitative results are shown in Table 2. Directly performing diffusion generation in high-resolution pixel space is highly challenging due to substantial computational costs and optimization difficulties, leading to significant performance degradation. As illustrated in Figure 6, the baseline model struggles with preserving global semantics and generating fine-grained visual details. In contrast, the proposed Semantics-Prompted DiT (SP-DiT) addresses these challenges, achieving significantly improved accuracy, for example, a 78% gain on the NYUv2 AbsRel metric. We further introduce a novel Cascaded DiT Design (Cas-DiT) that progressively increases the number of tokens. This coarse-to-fine design not only significantly improves efficiency, for example, reducing inference time by 30% on an RTX 4090 GPU, but also better models global context, leading to noticeable gains in accuracy.

Ablations on vision foundation models (VFMs). We evaluate the performance of SP-DiT using pretrained vision encoders from different VFMs, including MAE [24], DINOv2 [44], Depth Anything v2 [82], and VGGT [65], as illustrated in Table 3. All of them significantly boost performance.

4.3 Zero-Shot Relative Depth Estimation

To evaluate our model's zero-shot generalization, we compare it with recent depth estimation models [82, 4, 34, 23, 20] on five real-world benchmarks. As shown in Table 1, our model outperforms all other generative depth estimation models for all evaluation metrics. Unlike previous generative models, we do not rely on image priors from a pretrained Stable Diffusion [51] model. Instead, our diffusion model is trained from scratch and still achieves superior performance. Our model generalizes well to a wide range of real-world scenes, even when trained solely on synthetic depth datasets. Visual comparisons are shown in Figure 4, our model (1024) preserves more fine-grained details than Depth Anything v2 [82] and MoGe 2 [70]. Moreover, it demonstrates significantly higher robustness than Depth Pro [4], especially in challenging regions with complex textures, cluttered backgrounds, or large sky areas.

4.4 Edge-Aware Point Cloud Evaluation

Our objective is to estimate pixel-perfect depth maps that yield clean point clouds without *flying pixels*, which often occur at object edges due to inaccurate depth predictions in these regions. However, existing evaluation benchmarks and metrics often struggle to reflect *flying pixels* at object edges. For

Table 3: **Ablation studies on Vision Foundation Models (VFMs).** Note that we only utilize a pretrained encoder from these VFMs, such as a 24-layer ViT from DINOv2 or Depth Anything v2.

VFM Type	NYUv2		KITTI		ETH3D		ScanNet		DIODE	
	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑
w/o SP-DiT	22.5	72.8	27.3	63.9	12.1	87.4	25.7	65.1	23.9	76.5
SP-DiT (MAE [24])	6.4	95.0	14.4	84.9	7.3	94.8	7.7	92.5	11.6	91.3
SP-DiT (DINOv2 [44])	4.8	96.4	9.3	91.2	5.6	96.2	5.1	96.9	9.2	93.5
SP-DiT (VGGT [65])	4.7	96.7	**7.6**	**94.1**	**4.1**	**97.8**	**3.8**	**98.0**	7.8	94.9
SP-DiT (DepthAny. v2 [82])	**4.3**	**97.4**	8.0	93.1	4.5	97.7	4.5	97.3	**7.0**	**95.5**

Table 4: **Edge-aware point cloud evaluation.** Our model achieves the best performance on the high-quality Hypersim test set. To further verify that VAE compression leads to *flying pixels*, we evaluate the ground truth depth maps after VAE reconstruction, denoted as GT(VAE).

	Marigold[34]	GeoWizard[17]	DepthAny. v2[82]	DepthPro[4]	GT(VAE)	Ours
Chamfer Dist.↓	0.17	0.16	0.18	0.14	0.12	**0.08**

example, benchmarks like NYUv2 or KITTI usually lack edge annotations, while metrics such as AbsRel and δ_1 are dominated by flat regions, making it difficult to assess depth accuracy at edges.

To address these limitations, we evaluate on the official test split of the Hypersim [50] dataset, which provides high-quality ground-truth point clouds and is not used during training. We further propose an edge-aware point cloud metric that quantifies depth accuracy at edges. Specifically, we extract edge masks from ground-truth depth maps using the Canny operator and compute the Chamfer Distance between predicted and ground-truth point clouds near these edges.

Quantitative results in Table 4 show that our method achieves the best performance. Discriminative models like Depth Pro [4] and Depth Anything v2 [82] tend to smooth edges, causing *flying pixels*. Generative models such as Marigold [34] rely on VAE compression, which blurs edges and details, causing artifacts in the reconstructed point clouds. To illustrate this, we encode and decode the ground-truth depth using a VAE (GT(VAE)), without any generative process. Table 4 and Figure 2 show that VAE compression introduces *flying pixels*, leading to a larger Chamfer Distance than ours.

5 Conclusion

We presented **Pixel-Perfect Depth**, a monocular depth estimation model that leverages pixel-space diffusion transformers to produce high-quality, flying-pixel-free point clouds. Unlike prior generative depth models that rely on latent-space diffusion with a VAE, our model performs diffusion directly in the pixel space, avoiding *flying pixels* caused by VAE compression. To tackle the complexity and optimization challenges of pixel-space diffusion, we introduce Semantics-Prompted DiT and Cascade DiT Design, which greatly boost performance. Our model significantly outperforms prior models in edge-aware point cloud evaluation.

Limitations and future work. This work has two known limitations. First, like most image-based diffusion models, it lacks temporal consistency when applied to video frames, resulting in a little flickering depth across frames. Second, its multi-step diffusion process leads to slower inference compared to discriminative models like Depth Anything v2. Future works can address these limitations by exploring video depth estimation methods [57, 31, 80, 6, 33] to improve temporal consistency and adopting DiT acceleration strategies to speed up inference.

Acknowledgements. This research is supported by the National Key R&D Program of China (2024YFE0217700), the National Natural Science Foundation of China (623B2036, 62472184), the Fundamental Research Funds for the Central Universities, and the Innovation Project of Optics Valley Laboratory (Grant No. OVL2025YZ005).

References

[1] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

[2] Yunpeng Bai and Qixing Huang. Fiffdepth: Feed-forward transformation of diffusion-based generators for detailed depth estimation. *arXiv preprint arXiv:2412.00671*, 2024.

[3] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv*, 2023.

[4] Aleksei Bochkovskii, AmaÃĢl Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024.

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

[6] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. *arXiv preprint arXiv:2501.12375*, 2025.

[7] Junda Cheng, Wenjing Liao, Zhipeng Cai, Longliang Liu, Gangwei Xu, Xianqi Wang, Yuzhou Wang, Zikang Yuan, Yong Deng, Jinliang Zang, Yangyang Shi, Jinhui Tang, and Xin Yang. Monster++: Unified stereo matching, multi-view stereo, and real-time stereo with monodepth priors. *arXiv preprint arXiv:2501.08643*, 2025.

[8] Junda Cheng, Longliang Liu, Gangwei Xu, Xianqi Wang, Zhaoxing Zhang, Yong Deng, Jinliang Zang, Yurui Chen, Zhipeng Cai, and Xin Yang. Monster: Marry monodepth to stereo unleashes power. *CVPR*, 2025.

[9] Junda Cheng, Wei Yin, Kaixuan Wang, Xiaozhi Chen, Shijie Wang, and Xin Yang. Adaptive fusion of single-view and multi-view depth for autonomous driving. In *CVPR*, pages 10138–10147, 2024.

[10] Jaehoon Cho, Dongbo Min, Youngjung Kim, and Kwanghoon Sohn. Diml/cvl rgb-d dataset: 2m rgb-d images of natural indoor and outdoor scenes. *arXiv*, 2021.

[11] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017.

[12] Yiquan Duan, Xianda Guo, and Zheng Zhu. Diffusiondepth: Diffusion denoising approach for monocular depth estimation. In *ECCV*, pages 432–449. Springer, 2024.

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

[14] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, pages 2650–2658, 2015.

[15] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 27, 2014.

[16] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

[17] 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. In *ECCV*, pages 241–258. Springer, 2025.

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361. IEEE, 2012.

[19] Jose L Gómez, Manuel Silva, Antonio Seoane, Agnès Borrás, Mario Noriega, Germán Ros, Jose A Iglesias-Guitian, and Antonio M López. All for one, and one for all: Urbansyn dataset, the third musketeer of synthetic driving scenes. *Neurocomputing*, 637:130038, 2025.

[20] Ming Gui, Johannes Schusterbauer, Ulrich Prestel, Pingchuan Ma, Dmytro Kotovenko, Olga Grebenkova, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Depthfm: Fast generative monocular depth estimation with flow matching. In *AAAI*, volume 39, pages 3203–3211, 2025.

[21] Vitor Guizilini, Igor Vasiljevic, Dian Chen, Rareș Ambruș, and Adrien Gaidon. Towards zero-shot scale-aware monocular depth estimation. In *ICCV*, pages 9233–9243, 2023.

[22] Xianda Guo, Chenming Zhang, Juntao Lu, Yiqi Wang, Yiqun Duan, Tian Yang, Zheng Zhu, and Long Chen. Openstereo: A comprehensive benchmark for stereo matching and strong baseline. *arXiv preprint arXiv:2312.00343*, 2023.

[23] Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Liu, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv*, 2024.

[24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022.

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NIPS*, 33:6840–6851, 2020.

[26] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *JMLR*, 23(47):1–33, 2022.

[27] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[28] Derek Hoiem, Alexei A Efros, and Martial Hebert. Recovering surface layout from an image. *IJCV*, 75:151–172, 2007.

[29] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *ICML*, pages 13213–13232. PMLR, 2023.

[30] 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. *TPAMI*, 2024.

[31] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. *arXiv preprint arXiv:2409.02095*, 2024.

[32] 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.

[33] Bingxin Ke, Dominik Narnhofer, Shengyu Huang, Lei Ke, Torben Peters, Katerina Fragkiadaki, Anton Obukhov, and Konrad Schindler. Video depth without video models. *arXiv preprint arXiv:2411.19189*, 2024.

[34] 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*, pages 9492–9502, 2024.

[35] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[36] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018.

[37] Zhenyu Li, Shariq Farooq Bhat, and Peter Wonka. Patchfusion: An end-to-end tile-based framework for high-resolution monocular metric depth estimation. In *CVPR*, pages 10016–10025, 2024.

[38] Zhenyu Li, Shariq Farooq Bhat, and Peter Wonka. Patchrefiner: Leveraging synthetic data for real-domain high-resolution monocular metric depth estimation. In *ECCV*, pages 250–267. Springer, 2024.

[39] Dingkang Liang, Tianrui Feng, Xin Zhou, Yumeng Zhang, Zhikang Zou, and Xiang Bai. Parameter-efficient fine-tuning in spectral domain for point cloud learning. *TPAMI*, 2025.

[40] Dingkang Liang, Wei Hua, Chunsheng Shi, Zhikang Zou, Xiaoqing Ye, and Xiang Bai. Sood++: Leveraging unlabeled data to boost oriented object detection. *TPAMI*, 2025.

[41] Haotong Lin, Sida Peng, Jingxiao Chen, Songyou Peng, Jiaming Sun, Minghuan Liu, Hujun Bao, Jiashi Feng, Xiaowei Zhou, and Bingyi Kang. Prompting depth anything for 4k resolution accurate metric depth estimation. In *CVPR*, pages 17070–17080, 2025.

[42] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[43] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[44] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[45] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[46] 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*, 2024.

[47] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

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

[49] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020.

[50] 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.

[51] 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.

[52] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *TPAMI*, 31(5):824–840, 2008.

[53] 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. *NIPS*, 36:39443–39469, 2023.

[54] Saurabh Saxena, Junhwa Hur, Charles Herrmann, Deqing Sun, and David J Fleet. Zero-shot metric depth with a field-of-view conditioned diffusion model. *arXiv preprint arXiv:2312.13252*, 2023.

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

[56] 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.

[57] Jiahao Shao, Yuanbo Yang, Hongyu Zhou, Youmin Zhang, Yujun Shen, Vitor Guizilini, Yue Wang, Matteo Poggi, and Yiyi Liao. Learning temporally consistent video depth from video diffusion priors. *arXiv preprint arXiv:2406.01493*, 2024.

[58] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760. Springer, 2012.

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

[60] Ziyang Song, Zerong Wang, Bo Li, Hao Zhang, Ruijie Zhu, Li Liu, Peng-Tao Jiang, and Tianzhu Zhang. Depthmaster: Taming diffusion models for monocular depth estimation. *arXiv preprint arXiv:2501.02576*, 2025.

[61] Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. *arXiv preprint arXiv:2309.03350*, 2023.

[62] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *CVPR*, pages 8942–8952, 2021.

[63] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019.

[64] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes. In *3DV*, 2019.

[65] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. *arXiv preprint arXiv:2503.11651*, 2025.

[66] Jiyuan Wang, Chunyu Lin, Cheng Guan, Lang Nie, Jing He, Haodong Li, Kang Liao, and Yao Zhao. Jasmine: Harnessing diffusion prior for self-supervised depth estimation. *arXiv preprint arXiv:2503.15905*, 2025.

[67] JiYuan Wang, Chunyu Lin, Lei Sun, Rongying Liu, Lang Nie, Mingxing Li, Kang Liao, Xiangxiang Chu, and Yao Zhao. From editor to dense geometry estimator. *arXiv preprint arXiv:2509.04338*, 2025.

[68] Qiang Wang, Shizhen Zheng, Qingsong Yan, Fei Deng, Kaiyong Zhao, and Xiaowen Chu. Irs: A large naturalistic indoor robotics stereo dataset to train deep models for disparity and surface normal estimation. In *ICME*, 2021.

[69] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *CVPR*, pages 5261–5271, 2025.

[70] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025.

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

[72] Xianqi Wang, Gangwei Xu, Hao Jia, and Xin Yang. Selective-stereo: Adaptive frequency information selection for stereo matching. In *CVPR*, pages 19701–19710, 2024.

[73] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruibo Li, and Zhenbo Luo. Monocular relative depth perception with web stereo data supervision. In *CVPR*, 2018.

[74] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. Structure-guided ranking loss for single image depth prediction. In *CVPR*, 2020.

[75] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. Iterative geometry encoding volume for stereo matching. In *CVPR*, pages 21919–21928, 2023.

[76] Gangwei Xu, Xianqi Wang, Zhaoxing Zhang, Junda Cheng, Chunyuan Liao, and Xin Yang. Igev++: Iterative multi-range geometry encoding volumes for stereo matching. *TPAMI*, 2025.

[77] Gangwei Xu, Yun Wang, Junda Cheng, Jinhui Tang, and Xin Yang. Accurate and efficient stereo matching via attention concatenation volume. *TPAMI*, 2023.

[78] Guangkai Xu, Yongtao Ge, Mingyu Liu, Chengxiang Fan, Kangyang Xie, Zhiyue Zhao, Hao Chen, and Chunhua Shen. What matters when repurposing diffusion models for general dense perception tasks? *arXiv preprint arXiv:2403.06090*, 2024.

[79] Tian-Xing Xu, Xiangjun Gao, Wenbo Hu, Xiaoyu Li, Song-Hai Zhang, and Ying Shan. Geometrycrafter: Consistent geometry estimation for open-world videos with diffusion priors. *arXiv preprint arXiv:2504.01016*, 2025.

[80] Honghui Yang, Di Huang, Wei Yin, Chunhua Shen, Haifeng Liu, Xiaofei He, Binbin Lin, Wanli Ouyang, and Tong He. Depth any video with scalable synthetic data. *arXiv*, 2024.

[81] 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*, pages 10371–10381, 2024.

[82] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *NIPS*, 37:21875–21911, 2024.

[83] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[84] Jingfeng Yao, Cheng Wang, Wenyu Liu, and Xinggang Wang. Fasterdit: Towards faster diffusion transformers training without architecture modification. *NIPS*, 37:56166–56189, 2024.

[85] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *arXiv preprint arXiv:2501.01423*, 2025.

[86] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *CVPR*, 2020.

[87] 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.

[88] 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 *CVPR*, pages 9043–9053, 2023.

[89] 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.

[90] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

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

[92] Xiang Zhang, Bingxin Ke, Hayko Riemenschneider, Nando Metzger, Anton Obukhov, Markus Gross, Konrad Schindler, and Christopher Schroers. Betterdepth: Plug-and-play diffusion refiner for zero-shot monocular depth estimation. *arXiv preprint arXiv:2407.17952*, 2024.

[93] 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.

[94] Wendi Zheng, Jiayan Teng, Zhuoyi Yang, Weihan Wang, Jidong Chen, Xiaotao Gu, Yuxiao Dong, Ming Ding, and Jie Tang. Cogview3: Finer and faster text-to-image generation via relay diffusion. In *ECCV*, pages 1–22. Springer, 2024.

[95] Lianghui Zhu, Zilong Huang, Bencheng Liao, Jun Hao Liew, Hanshu Yan, Jiashi Feng, and Xinggang Wang. Dig: Scalable and efficient diffusion models with gated linear attention. *arXiv preprint arXiv:2405.18428*, 2024.

[96] Lianghui Zhu, Zilong Huang, Bencheng Liao, Jun Hao Liew, Hanshu Yan, Jiashi Feng, and Xinggang Wang. Dig: Scalable and efficient diffusion models with gated linear attention. In *CVPR*, pages 7664–7674, 2025.

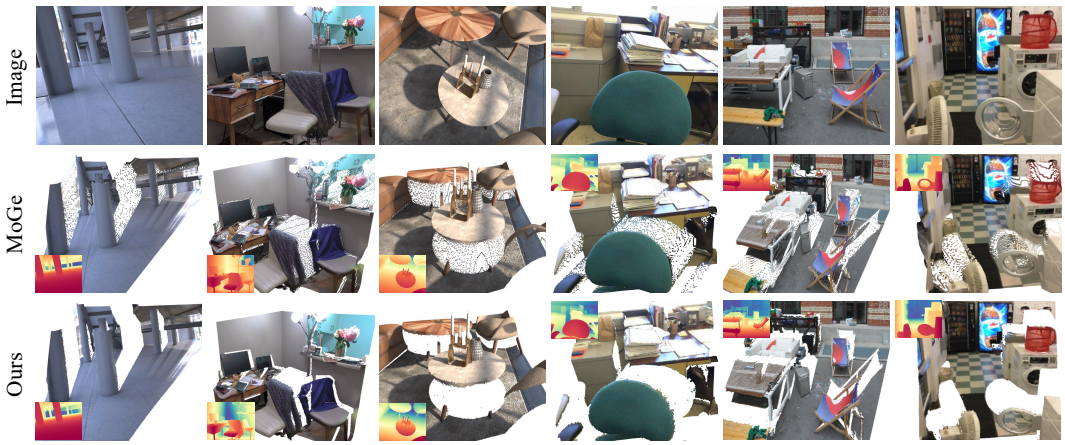

Figure 7: **Qualitative comparisons with MoGe [69].** Top: input images are taken from four test sets: Hypersim [50], DIODE [63], ScanNet [11], and ETH3D [56]. Middle: results of MoGe [69]. Bottom: our results. As a discriminative model, MoGe [69], like other discriminative models [82, 4], also suffers from *flying pixels* at edges and details.

Table 5: **Quantitative comparisons with REPA [90].** Our model significantly outperforms REPA [90]. To ensure a fair comparison, the pretrained vision encoder used in both DiT+REPA and DiT+Ours is kept the same.

Method	NYUv2		KITTI		ETH3D		ScanNet		DIODE	
	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑
DiT (vanilla)	22.5	72.8	27.3	63.9	12.1	87.4	25.7	65.1	23.9	76.5
DiT+REPA [90]	17.6	78.0	23.4	70.6	9.1	91.2	20.1	74.3	14.6	86.9
DiT+Ours	**4.3**	**97.4**	**8.0**	**93.1**	**4.5**	**97.7**	**4.5**	**97.3**	**7.0**	**95.5**

A Qualitative Comparisons with MoGe

We provide qualitative comparisons of reconstructed point clouds, as shown in Figure 7. MoGe [69], as a discriminative model, suffers from *flying pixels* at edges and fine structures, a common issue observed in other discriminative models [82, 4]. Our model produces significantly fewer *flying pixels* compared to MoGe [69].

B Additional Discussion with REPA

We provide an additional discussion on the recent image generation method REPA [90]. REPA [90] aligns intermediate tokens in diffusion models with pretrained vision encoder, significantly improving training efficiency and generation quality for image generation tasks. We compare our method with REPA [90], and the quantitative evaluation results are presented in Table 5. DiT+REPA refers to training the DiT model with REPA's representation alignment, while DiT+Ours denotes training the DiT model using our Semantics-Prompted DiT. For a fair comparison, the pretrained vision encoder used in both DiT+REPA and DiT+Ours is kept the same. Experimental results show that our Semantics-Prompted DiT significantly outperforms REPA [90]. We attribute our model's superiority over REPA to two factors. First, during training, REPA's implicit alignment of DiT tokens with the pretrained vision encoder is suboptimal, making it difficult for DiT to effectively leverage semantic prompts from the pretrained vision encoder. In contrast, our Semantics-Prompted DiT directly integrates semantic cues, resulting in more effective prompts. Second, at inference, REPA cannot leverage the pretrained vision encoder to provide semantic prompts, whereas our method effectively incorporates high-level semantics into the Semantics-Prompted DiT during inference to prompt the diffusion process.

Table 6: **Runtime comparison on RTX 4090 GPU.** The runtime is measured using the 512×512 model with 4 denoising steps.

	Depth Anything v2 [82]	DepthPro [4]	PPD-Large	PPD-Small
Time (ms)	18	170	140	40

Table 7: **Quantitative comparisons between PPD-Large and PPD-Small.**

Method	NYUv2		KITTI		ETH3D		ScanNet		DIODE	
	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑	AbsRel↓	δ_1↑
PPD-Small	4.5	97.3	8.3	92.8	4.6	97.4	4.7	97.2	7.3	95.3
PPD-Large	**4.3**	**97.4**	**8.0**	**93.1**	**4.5**	**97.7**	**4.5**	**97.3**	**7.0**	**95.5**

C Analysis of Flying Pixels in Different Types of VAEs

To better understand the emergence of *flying pixels* in VAE-based reconstructions, we analyze VAEs with different latent dimensions (*i.e.*, channel) by using them to reconstruct ground truth depth maps. Figure 8 shows that both VAE variants exhibit *flying pixels* at object edges and details, revealing a common weakness of VAE reconstructions in preserving precise geometric structures. VAE-d4 (SD2) denotes the reconstruction of ground truth depth maps using the VAE from Stable Diffusion 2, with a latent dimension of 4, which is also used in Marigold [34]. VAE-d16 (SD3.5) uses the VAE from Stable Diffusion 3.5, which has a latent dimension of 16.

D Efficiency and Lightweight Variant

Our Pixel-Perfect Depth (PPD) model is slower than Depth Anything v2 [82] owing to the multi-step diffusion process, but its inference time remains comparable to Depth Pro [4], as shown in Table 6. To further accelerate inference, we develop a lightweight variant, PPD-Small, which achieves substantially faster runtime with only marginal accuracy loss, as shown in Table 7. In contrast to PPD-Large, PPD-Small is built upon DiT-Small with a reduced number of parameters, making it more suitable for efficient inference.

Image VAE-d4 (SD2) VAE-d16 (SD3.5) Ours

Figure 8: **Validation of flying pixels in different types of VAEs.** We present further qualitative comparisons showing that increasing the latent dimension in VAEs fails to eliminate *flying pixels*. VAE-d4 (SD2) denotes the reconstruction of ground truth depth maps using the VAE from Stable Diffusion 2, with a latent dimension of 4, which is also used in Marigold. VAE-d16 (SD3.5) uses the VAE from Stable Diffusion 3.5, which has a latent dimension of 16.

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 abstract and introduction clearly present the core contributions of the paper, which are subsequently supported by experimental results in the main body.

   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: We discuss the limitations of our model in the Conclusion section.

   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 contain formal theoretical results or proofs. The focus is on the design, implementation, and empirical evaluation of the proposed method.

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: The paper provides sufficient implementation details to reproduce the main experimental results, including model architecture, training settings, datasets, evaluation metrics, and ablation studies. We also plan to release the code and models to further facilitate reproducibility.

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: The datasets used in our experiments are publicly available, ensuring accessibility of the data. Although the code is not provided at submission time, we plan to release the code, models, and detailed instructions to facilitate full reproducibility.

   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: The paper provides comprehensive details on the experimental setup, including dataset splits, hyperparameter settings and optimizer type.

   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: [No]

   Justification: We report quantitative metrics for the depth estimation task and provide qualitative visualizations of the depth map and point clouds to demonstrate the effectiveness of our method. However, the current experiments do not include error bars or formal statistical significance analysis.

   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: We report the number of GPUs used during training. We provide the inference time measured on an RTX 4090 GPU.

   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: We have reviewed the NeurIPS Code of Ethics and confirm our research fully complies with its principles. No ethical issues arise.

   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: [NA]

Justification: This work primarily focuses on fundamental algorithm/model improvements and technical methods. There are currently no direct or significant societal impacts associated with the research; therefore, no discussion on positive or negative societal impacts is included.

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 does not involve the release of models or datasets that pose a high risk of misuse. Therefore, no specific safeguards are necessary.

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: We use publicly available datasets and pretrained models, all of which are properly cited in the paper.

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: Although the assets (code and model weights) are not released at submission time, we plan to release them upon acceptance. The released assets will be properly documented, including usage instructions, license information, and known limitations. No new datasets are introduced in this work.

    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: This paper does not involve crowdsourcing experiments or 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: This paper does not involve research with human subjects or crowdsourcing, and therefore no IRB or equivalent ethical approval is required.

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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method of this research does not involve any important, original, or non-standard usage of large language models (LLMs). Any use of LLMs was limited to writing assistance and does not impact the scientific contributions of the paper.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.