EAGLEDEPTH: EFFICIENT FINE-GRAINED DEPTH ESTIMATION VIA PIXEL DIFFUSION DECODER

Anonymous authors
Paper under double-blind review

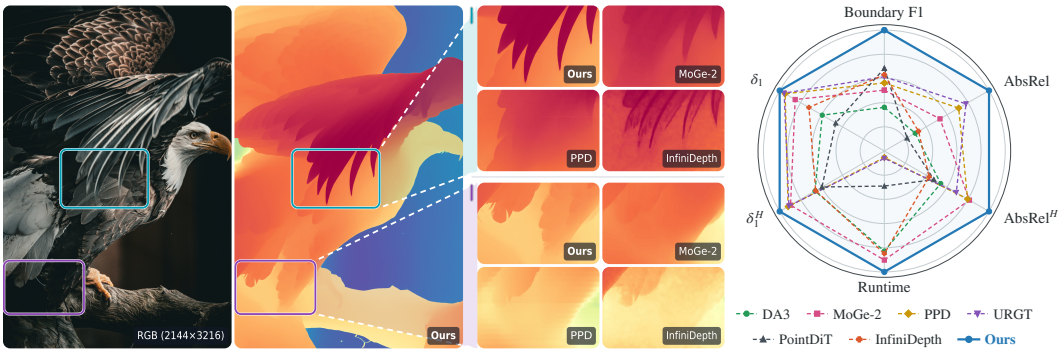


Figure 1: **EAGLEDEPTH excels at predicting fine-grained depth maps from high-resolution images.** Compared with state-of-the-art methods, our model achieves better depth estimation results with faster inference, as shown in the radar chart (quality metrics averaged over the five Synth4K (Yu et al., 2026a) subsets; larger radii indicate better performance).

ABSTRACT

Recovering detailed geometry from high-resolution images is critical for precise perception of the surroundings and objects. However, existing methods which use latent-space modeling and VAE reconstruction can compromise geometric details. Furthermore, decoding from latent codes introduces substantial inference overhead. To address those issues, we present EAGLEDEPTH, an efficient framework for high-resolution monocular depth estimation that combines the geometric priors of latent diffusion with fine-grained pixel-space generation. Our key idea is to retain depth-aware latent representations as guidance while generating the final depth map directly in pixel space. We train the latent and pixel components sequentially: first, we fine-tune a pretrained latent diffusion model using paired RGB–depth supervision; then, we adapt a pretrained pixel diffusion decoder, PiD, to predict depth conditioned on the learned features. Training of the pixel component starts at 1024 resolution and continues across multiple resolutions up to 4K. The latent branch processes resized, lower-resolution RGB images, while the pixel branch generates depth at the target resolution, bypassing the original VAE decoder. This design preserves learned geometric knowledge without requiring the latent backbone to operate at the output resolution. On five commonly used depth estimation datasets and the high-resolution Synth4K dataset, our framework achieves state-of-the-art depth estimation performance, with faster inference and better preservation of fine structures and object boundaries.

1 INTRODUCTION

Monocular depth estimation provides a fundamental capability for robotic navigation (Redder et al., 2025; Gode et al., 2025), manipulation (Zhang et al., 2026c;b; Yuan et al., 2025), and 3D scene understanding (Eigen et al., 2014; Ouyang et al., 2024; Tang et al., 2024). Modern cameras capture multi-megapixel images containing small objects, thin structures, and closely spaced boundaries, creating a demand for depth predictions that preserve geometric detail at comparable resolutions. However, representative generative depth and geometry estimators are trained at resolutions such as

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

512×512 and 1024×768 pixels (Xu et al., 2026a;b). Downsampling images to these working resolutions can weaken or eliminate visual cues associated with small structures. Efficiently recovering high-resolution, fine-grained geometry is therefore important for applications such as small-obstacle avoidance and precise robotic manipulation.

Preserving local detail and global scene consistency at high resolution requires careful control of computational cost. Interpolation cannot recover unresolved structures, while increasing the working resolution raises computation and memory demands. Methods such as PatchFusion (Li et al., 2024) and PatchRefiner (Li et al., 2026) refine local patches using global image context to recover detailed, coherent depth maps. For diffusion models, iterative denoising can further increase inference costs, particularly when processing large spatial grids.

Generative methods (Hu et al., 2025; Fu et al., 2024) such as Marigold (Ke et al., 2024b) demonstrate that pretrained latent diffusion models can acquire depth-specific knowledge through paired RGB–depth supervision. These representations help resolve ambiguity by providing learned geometric context when visual evidence is incomplete. However, latent compression can obscure narrow structures and closely spaced boundaries, while reconstruction with a variational autoencoder (VAE) can introduce additional geometric errors. Boundary degradation occurs even when reconstructing encoded ground-truth geometry (Xu et al., 2026a;b). Moreover, the decoder must reconstruct the full-resolution output, adding computation as image size increases. These limitations motivate retaining useful latent priors without the original reconstruction pathway.

Pixel-space diffusion avoids VAE reconstruction by generating geometric outputs directly. Pixel-Perfect Depth (Xu et al., 2026a) predicts depth using pretrained image encoders such as Depth Anything V2, while PointDiT (Xu et al., 2026b) generates dense 3D point maps conditioned on frozen DINOv3 image features. However, processing high-resolution spatial representations remains computationally demanding as the number of spatial tokens increases, and iterative sampling can further increase latency. Direct pixel-space generation therefore still faces the challenge of recovering fine geometry efficiently at multi-megapixel resolutions.

We introduce EAGLEDEPTH, a framework that combines depth-adapted latent representations with direct pixel-space prediction. Our key idea is to separate geometric prior extraction from high-resolution detail recovery to allocate computation across different resolutions. The latent branch supplies depth-aware guidance at low resolution, while the original RGB image provides the local evidence needed to resolve fine structures. We adapt the pretrained Pixel Diffusion Decoder (PiD) (Lu et al., 2026), originally designed to jointly decode and upsample image latents. Depth-specific supervision adapts this formulation to geometric prediction, where details must reflect meaningful surfaces and depth discontinuities rather than visual sharpness alone.

We train the two components sequentially. First, we fine-tune FLUX.2-klein-4B (Black Forest Labs, 2026), a pretrained latent diffusion model, on paired RGB–depth data. This branch processes resized, lower-resolution RGB images to provide depth-aware guidance. We then freeze this branch and fine-tune PiD, conditioned on its guidance and the original RGB image. Pixel training starts with approximately 1024^2 pixels per image and continues across multiple resolutions up to 4K. The pixel branch directly generates target-resolution depth, bypassing the original VAE decoder during inference without scaling the latent backbone to the output resolution.

We evaluate this design on six benchmark datasets, including the high-resolution Synth4K dataset, covering standard depth estimation and high-resolution prediction up to 4K. EAGLEDEPTH achieves state-of-the-art depth estimation performance, with better preservation of fine structures and object boundaries, together with faster inference.

Our contributions can be summarized as follows:

- We introduce EAGLEDEPTH, a generative monocular depth estimator that brings together the geometric knowledge of depth-adapted latent diffusion and the pixel-space modeling capabilities for efficient, detailed prediction at high resolutions.
- We show that depth-aware features from downsampled RGB inputs can guide depth generation at higher resolutions. This enables our framework to retain latent geometric priors without executing the latent backbone at the output resolution or using its VAE decoder.
- Experiments on five commonly used depth estimation datasets and the high-resolution Synth4K dataset establish state-of-the-art depth estimation results for our 1024-resolution

model and its multi-resolution extension supporting resolutions up to 4K, with consistent improvements in fine-structure recovery and boundary fidelity, alongside faster inference.

2 Related Work

2.1 Monocular Depth Estimation

Discriminative approaches such as DPT (Ranftl et al., 2021) and Depth Anything V2 (Yang et al., 2024) directly predict depth using pretrained visual representations, while PatchFusion (Li et al., 2024) extends depth estimation to high resolutions through patch-based fusion. Generative approaches leverage priors learned from image synthesis: Marigold (Ke et al., 2024b) and Lotus-G (He et al., 2025b) adapt pretrained latent diffusion models for depth prediction. To address geometric artifacts associated with latent reconstruction, Pixel-Perfect Depth (Xu et al., 2026a) directly generates depth maps in pixel space, while PointDiT (Xu et al., 2026b) applies pixel-space diffusion to dense 3D point maps. These methods demonstrate the value of pretrained representations and direct geometric generation, although efficiently recovering fine-grained geometry at multi-megapixel resolutions remains challenging. Our approach combines these strengths by using depth-adapted features from a lower-resolution latent backbone to guide depth generation at the target resolution.

2.2 Diffusion Models

Latent diffusion models (Rombach et al., 2022; Zhou et al., 2022) reduce generative modeling costs by operating in compressed representations and reconstructing outputs through pretrained autoencoders. Although computationally effective, this formulation makes output fidelity dependent on latent compression and reconstruction. Recent advances revisit diffusion directly in pixel space. JiT (Li & He, 2026) demonstrates effective clean-image prediction with plain transformers, while PixelDiT (Yu et al., 2026b) combines patch-level and pixel-level processing for efficient pixel-space generation. Complementing these approaches, PiD (Lu et al., 2026) introduces a latent-conditioned pixel diffusion decoder that jointly performs decoding and spatial upsampling, replacing conventional VAE decoding. We build on this pretrained decoder and adapt it from image synthesis to monocular depth estimation. Specifically, depth-aware features learned through latent diffusion adaptation condition pixel-space depth generation, allowing us to reuse pretrained generative knowledge for efficient, high-resolution geometric prediction.

3 Method

3.1 Preliminaries: Flow Matching

Flow matching learns a time-dependent velocity field that transports a noise distribution to a target data distribution (Lipman et al., 2022). Given a clean sample $\boldsymbol{x}_0$ and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, linear interpolation defines the state and target velocity at time $t \in [0, 1]$:

$$\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}, \qquad \boldsymbol{u}_t = \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{\epsilon} - \boldsymbol{x}_0, \tag{1}$$

with $t = 0$ denoting the data endpoint and $t = 1$ the noise endpoint. The trainable model can parameterize the velocity field $v_\psi(\boldsymbol{x}_t, t, c)$ directly (v-pred), or express it through a clean-sample predictor f_ψ (x-pred) as $v_\psi(\boldsymbol{x}_t, t, c) = (\boldsymbol{x}_t - f_\psi(\boldsymbol{x}_t, t, c))/t$, where c denotes conditioning information. At inference, samples are generated by solving the ODE $\mathrm{d}\boldsymbol{x}_t/\mathrm{d}t = v_\psi(\boldsymbol{x}_t, t, c)$ from $t = 1$ to $t = 0$.

Pretrained diffusion models provide rich visual priors for depth estimation (Ke et al., 2024a). Previous studies (He et al., 2025a;b; Zhang et al., 2026a) have shown that generative depth estimation models benefit from using RGB images as inputs rather than noise and can produce accurate predictions in a single forward pass. In line with these findings, we adopt a single-step flow matching formulation that treats the RGB input $\boldsymbol{x}_{\mathrm{rgb}}$ as the state $\boldsymbol{x}_{t^\star}$ at a fixed timestep $t^\star \in (0, 1)$. The depth estimate is obtained through a single update:

$$\hat{\boldsymbol{x}}_{\mathrm{depth}} = \boldsymbol{x}_{\mathrm{rgb}} - t^\star v_\psi(\boldsymbol{x}_{\mathrm{rgb}}, t^\star, c). \tag{2}$$

The same formulation applies in latent space for the latent diffusion model, with $\boldsymbol{x}_{\mathrm{rgb}}$ replaced by the encoded RGB latent $\boldsymbol{z}_{\mathrm{rgb}}$ and the output representing a depth latent.

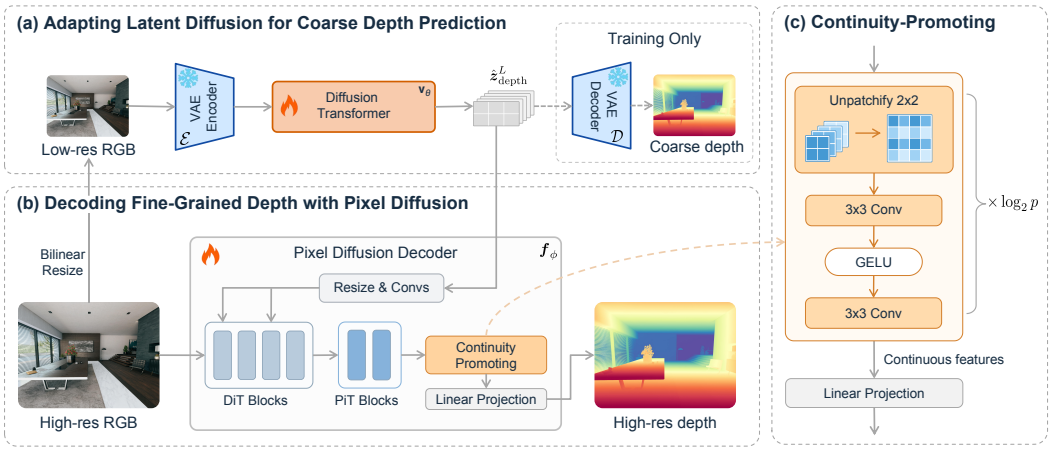


Figure 2: **Method overview.** **(a)** An adapted latent predictor, trained with a pixel-space loss, estimates a coarse depth latent from a downsampled RGB image. **(b)** Conditioned on this latent, the pixel diffusion decoder generates a high-resolution depth map with fine-grained details. **(c)** A lightweight continuity-promoting module mitigates discontinuity artifacts at patch boundaries.

3.2 Framework Overview

We aim to produce detailed, high-resolution depth maps, which requires capturing both scene-wide depth relationships and fine local geometry. Pretrained latent diffusion models combine rich visual priors with compact latent-space modeling. However, spatial compression and autoencoder reconstruction can lose fine details, and high-resolution VAE decoding incurs substantial computational and memory overhead. To address these limitations, we draw on the fine-detail modeling capabilities of pixel diffusion. Specifically, we adapt a pixel diffusion decoder pretrained for image synthesis, PiD (Lu et al., 2026), to decode coarse depth latents predicted by a depth-adapted latent diffusion model. During decoding, the pixel model uses high-resolution RGB observations to recover depth details beyond the coarse representation, producing a detailed depth map at the target resolution.

Figure 2 illustrates the overall framework of our proposed method. Given a high-resolution RGB image $\boldsymbol{x}^H_{\text{rgb}} \in \mathbb{R}^{H\times W\times 3}$, we resize it to a lower resolution (i.e., 512^2 pixels in total) and encode it using the pretrained autoencoder encoder $\mathcal{E}$. The adapted diffusion transformer v_θ predicts the latent velocity, which is used to obtain a coarse depth latent through a single flow update:

$$\boldsymbol{z}^L_{\text{rgb}} = \mathcal{E}\big(\mathcal{R}_L(\boldsymbol{x}^H_{\text{rgb}})\big)\,, \qquad \hat{\boldsymbol{z}}^L_{\text{depth}} = \boldsymbol{z}^L_{\text{rgb}} - t^\star v_\theta(\boldsymbol{z}^L_{\text{rgb}}, t^\star), \tag{3}$$

where $\mathcal{R}_L$ denotes low-resolution resizing, $t^\star$ is a fixed flow timestep ($t^\star = 0.5$ in our method), and $\boldsymbol{z}^L_{\text{rgb}}, \hat{\boldsymbol{z}}^L_{\text{depth}} \in \mathbb{R}^{h_L\times w_L\times C_z}$ denote the RGB and depth latents, respectively.

The pixel diffusion decoder then decodes the predicted coarse depth latent into a high-resolution depth map. Our pixel decoder backbone, PiD, builds on a pixel diffusion PixelDiT (Yu et al., 2026b), which uses diffusion transformer (DiT) blocks to process patch-level features and pixel transformer (PiT) blocks to refine pixel-level details. PiD interpolates the latents to be decoded to the target patch grid of size $H/p \times W/p$ and injects them into the DiT blocks through a lightweight convolutional adapter and gated feature addition, where p denotes the patch size. We retain this conditioning architecture, replacing the image latent with the predicted coarse depth latent and using the original high-resolution RGB image $\boldsymbol{x}^H_{\text{rgb}}$ as the pixel diffusion input state. We further insert a continuity-promoting module before the final linear projection to reduce patch-boundary artifacts in the predicted depth. The resulting decoder predicts high-resolution depth in a single step:

$$\hat{\boldsymbol{x}}^H_{\text{depth}} = \Pi\big[f_\phi\big(\boldsymbol{x}^H_{\text{rgb}}, t^\star, \hat{\boldsymbol{z}}^L_{\text{depth}}\big)\big] \in \mathbb{R}^{H\times W}, \tag{4}$$

where f_ϕ is the pixel diffusion decoder and Π represents averaging the three output channels into a single depth map. This coarse-to-fine design enables efficient high-resolution depth prediction with accurate scene-level structure and fine geometric detail.

3.3 Adapting Latent Diffusion for Coarse Depth Prediction

Accurate coarse depth provides reliable geometric guidance for subsequent detail refinement, motivating us to improve the accuracy of the latent depth predictor. Prior works have established effective strategies for adapting pretrained generative models to depth estimation (Ke et al., 2024a; He et al., 2025a;b). Motivated by these findings, we adopt deterministic single-step prediction and replace noise inputs with encoded RGB images. We use FLUX.2-klein-4B (Black Forest Labs, 2026), a pretrained rectified-flow transformer, as our backbone to balance model capability and computational efficiency. To improve accuracy, we revisit the supervision objective for the latent predictor.

Conventional latent flow matching trains models to predict instantaneous velocities in latent space. However, flow trajectories are not necessarily straight, so even accurate instantaneous velocities may yield endpoint errors in a single-step update (Frans et al., 2025). Moreover, although latent-space supervision encourages predictions to match targets, the decoder may still amplify small residual errors. Thus, latent-space supervision may be suboptimal. Since our single-step formulation does not require iterative latent-space sampling, we can directly supervise the decoded depth map:

$$\hat{\boldsymbol{x}}^L_{\text{depth}} = \Pi\big[\mathcal{D}\big(\hat{\boldsymbol{z}}^L_{\text{depth}}\big)\big]\,, \qquad \mathcal{L}_{\text{coarse}} = \mathcal{L}_{\text{depth}}\big(\hat{\boldsymbol{x}}^L_{\text{depth}}, \boldsymbol{x}^L_{\text{depth}}\big)\,, \tag{5}$$

where $\mathcal{D}$ denotes the VAE decoder, Π averages the output channels, $\boldsymbol{x}^L_{\text{depth}}$ is the normalized low-resolution ground-truth depth, and $\mathcal{L}_{\text{depth}}$ denotes the loss used for depth supervision. We freeze the decoder and backpropagate the depth loss through it, training the latent predictor under a fixed decoding mapping. This objective evaluates the quality of latent predictions by the accuracy of their decoded depth maps, encouraging accurate depth values and geometric structure.

3.4 Decoding Fine-Grained Depth with Pixel Diffusion

Central to our approach is adapting a pretrained pixel diffusion decoder to depth estimation. Compared with a conventional VAE decoder trained for reconstruction, a pixel diffusion decoder provides a generative prior learned in pixel space that can help recover fine details beyond the coarse latent representation. Specifically, we build upon PiD (Lu et al., 2026), which formulates latent decoding as latent-conditioned image generation in pixel space.

While the original model decodes natural-image latents, we fine-tune it to decode the predicted coarse depth latents into detailed, high-resolution depth maps using the high-resolution RGB input, as described in Section 3.2. Following prior findings on pixel-space diffusion (Li & He, 2026), we use direct clean-sample prediction (x-pred). To initialize the x-pred decoder from PiD's pretrained velocity model, we negate the weights and bias of its final linear layer: the negated velocity target, $-\boldsymbol{u}_t = \boldsymbol{x}_0 - \boldsymbol{\epsilon}$, differs from the clean-sample target $\boldsymbol{x}_0$ only by a noise term.

The unpatchification head in diffusion transformers maps each patch token to a $p \times p$ pixel block without local spatial mixing, leaving discontinuities along patch boundaries. In depth maps, these artifacts can be especially noticeable at object boundaries, where one patch spans sharply different depths. Convolutions can mitigate these inconsistencies through local feature fusion (He et al., 2025a; Wang et al., 2026a). However, the large patch size in pixel diffusion makes it costly for convolutional heads to capture patch-wide context after unpatchification. We therefore insert a continuity-promoting module before the original output projection, interleaving progressive unpatchification with small-kernel convolutions to mix hidden features at multiple spatial scales.

Specifically, given the normalized hidden features $\boldsymbol{h}^{(0)}$, the module progressively unfolds their within-patch dimensions into the spatial grid and applies convolutions after each unfolding step:

$$\boldsymbol{h}^{(k+1)} = \mathrm{Conv}_{3\times3}\Big(\mathrm{GELU}\Big(\mathrm{Conv}_{3\times3}\Big(\mathrm{Unpatchify}_{2\times2}\Big(\boldsymbol{h}^{(k)}\Big)\Big)\Big)\Big)\,, \quad k = 0, \ldots, \log_2 p - 1. \tag{6}$$

Each step doubles the spatial grid dimensions until the full pixel resolution is reached. The original linear projection is then applied to produce the output. Each Conv–GELU–Conv block is jointly initialized to implement an identity mapping, preserving the pretrained readout at initialization.

3.5 Training Strategy

We train the two components sequentially. First, we adapt the latent diffusion model for coarse depth prediction at approximately 512^2 pixels, learning to provide reliable scene-level geometry for

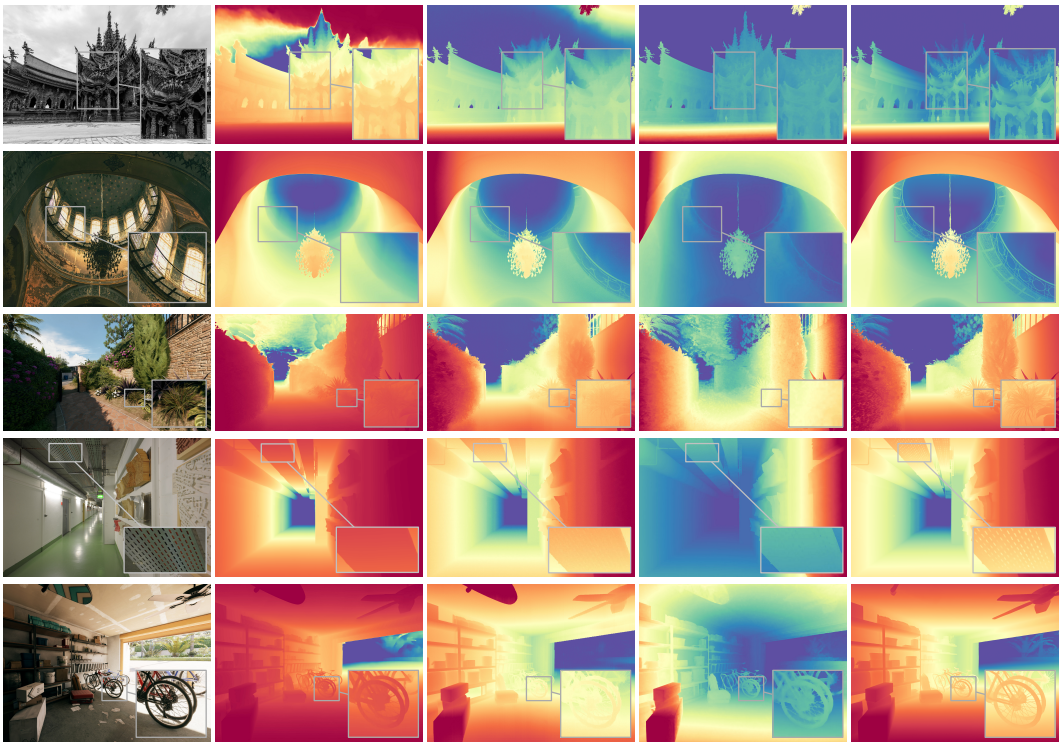


Figure 3: **Qualitative comparison with leading monocular depth estimation methods.** Our method preserves fine geometric details and thin structures while maintaining sharp depth boundaries and smooth surfaces, as highlighted by the arrows and zoomed-in regions.

subsequent detail refinement. We then freeze this model and fine-tune PiD at approximately 1024^2 pixels, adapting its pretrained generative prior to decode the predicted depth latents and recover fine geometric details with RGB guidance. This yields our Ours (1024) variant.

To further extend the model's high-resolution capability, we continue training PiD with mixed resolutions up to 4K. During this training, 25% of batches follow the same 1024^2 data settings as before, while the remaining 75% are drawn exclusively from high-resolution datasets across multiple resolution buckets. Retaining the 1024^2 training distribution helps preserve the model's existing capability as it learns to recover details at higher resolutions. This mixed-resolution training yields our Ours (MR) variant. Throughout PiD training, the latent model remains frozen and processes inputs at approximately 512^2 pixels regardless of PiD's output resolution, while its predicted latents are resized to the target patch grid before injection into PiD.

Both components are optimized using an L1 loss, a gradient matching loss (Yang et al., 2024), and a virtual normal loss (Yin et al., 2019), with respective weights of 1, 2, and 0.5.

4 Experiments

4.1 Experimental Setup

Datasets. Our model is trained on 536K RGB–depth pairs sampled from several datasets: Hypersim (Roberts et al., 2021), VKITTI2 (Cabon et al., 2020), TartanAir (Wang et al., 2020), IRS (Wang et al., 2021), EDEN (Le et al., 2021), RobbyReal (Tan et al., 2026), UnrealStereo4K (Zhang et al., 2018), UrbanSyn (Gómez et al., 2025), Spring (Mehl et al., 2023), and Rawmantic3D (Team, 2026). Only the last four datasets are used as high-resolution training dataset. We evaluate depth estimation performance on five real-world benchmarks: NYUv2 (Silberman et al., 2012), KITTI (Geiger et al., 2012), ETH3D (Schops et al., 2017), ScanNet (Dai et al., 2017), and DIODE (Vasiljevic et al., 2019). To further assess high-resolution prediction and fine-detail recovery, we use Synth4K (Yu et al., 2026a), a synthetic high-resolution benchmark that comprises five subsets.

Table 1: **Quantitative results on standard depth benchmarks. Bold** and underlined values denote the best and second-best results, respectively. Our method achieves the best results on four of the five benchmarks and remains competitive on KITTI.

Method	NYUv2		KITTI		ETH3D		ScanNet		DIODE	
	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑
MiDaS	11.1	88.5	23.6	63.0	18.4	75.2	12.1	84.6	—	—
DA2	4.5	97.9	7.4	94.6	13.1	86.5	6.5	97.2	6.6	95.2
Marigold	5.5	96.4	9.9	91.6	6.5	96.0	6.4	95.1	10.0	90.7
Lotus-2	5.0	97.6	6.6	95.7	5.1	97.5	5.2	97.3	6.1	95.2
DA3	4.1	98.0	5.9	97.2	4.4	97.8	3.9	97.9	5.9	95.1
MoGe-2	**3.6**	98.3	**5.2**	**97.9**	3.5	98.7	**3.5**	**98.3**	4.7	96.4
PPD	3.8	98.4	**5.2**	97.4	3.2	99.1	3.8	98.2	4.1	97.7
PointDiT	6.0	96.8	11.6	88.7	6.1	96.7	5.6	97.3	6.8	94.8
InfiniDepth	4.8	97.4	5.8	96.0	4.5	98.0	4.9	97.0	5.2	96.4
Ours (1024)	**3.6**	**98.5**	5.4	97.4	**3.1**	**99.2**	**3.5**	**98.3**	**3.9**	**97.9**
Ours (MR)	**3.6**	98.4	5.4	97.3	**3.1**	99.1	**3.5**	**98.3**	**3.9**	**97.9**

Implementation details. We transform the ground-truth depth into log space via $\log(1+d)$ and normalize each depth map to $-1, 1$ using its 2nd and 98th percentiles. We adapt FLUX.2-klein-4B (Labs, 2025) with rank-256 LoRA for 30,000 steps at a learning rate of 1×10^{-4}. With the latent model frozen, we fully fine-tune PiD (Lu et al., 2026) at approximately 1024^2 pixels for 50,000 steps with a learning rate of 3×10^{-5}, obtaining Ours (1024). Mixed-resolution training continues for another 20,000 steps with the learning rate reduced to 10^{-5}, producing Ours (MR). All models are trained on 8 NVIDIA GPUs, using the AdamW optimizer and a total batch size of 16.

Evaluation protocol. We evaluate all comparison methods under the same resolution settings for fair comparisons. Specifically, images from the five real-world benchmarks are resized to approximately 1024^2 pixels with aspect ratios preserved, whereas Synth4K is evaluated at native 4K resolution. We report AbsRel and δ_1 to assess overall depth accuracy. For datasets with dense ground-truth depth, we further report HF-masked AbsRel and δ_1 (Yu et al., 2026a) and boundary F1-score ($r = 1$) (Bochkovskiy et al., 2025) to assess fine-detail recovery in the predicted depth maps.

4.2 Comparison with State-of-the-Art Methods

We compare EagleDepth with several leading methods for depth estimation, including: MiDaS (Ranftl et al., 2020), DA2 (Yang et al., 2024), Marigold (Ke et al., 2024b), Lotus-2 (He et al., 2025a), DA3 (Lin et al., 2025), MoGe-2 (Wang et al., 2026b), PPD (Xu et al., 2026a), PointDiT (Xu et al., 2026b), InfiniDepth (Yu et al., 2026a), and URGT (Cui et al., 2026), where the last two methods are designed for high-resolution depth prediction.

Qualitative results. We present visual comparisons across diverse scenes and image resolutions in Figure 3. EagleDepth consistently captures the overall scene structure while preserving local geometric details. Compared with other methods, our model more faithfully recovers fine structures and delineates object boundaries that are otherwise blurred or lost. These visual improvements agree with the gains in HF-masked accuracy and boundary F1, supporting the effectiveness of combining coarse depth guidance with RGB-guided pixel decoding across scenes and resolutions.

Quantitative results. Table 1 reports depth accuracy on five real-world benchmarks. EagleDepth achieves competitive performance across these datasets, with both variants outperforming the comparison methods in AbsRel and δ_1 on ETH3D and DIODE. Ours (MR) maintains accuracy comparable to Ours (1024), indicating that mixed-resolution continuation largely preserves performance. On Synth4K, evaluated at native 4K resolution, Ours (MR) achieves strong overall accuracy and leads all comparison methods in both HF-masked metrics and boundary F1 across all five subsets (Tables 2 and 3). Meanwhile, the Ours (1024) model, which is trained at a lower resolution like most comparison methods, remains competitive in overall accuracy and fine-detail recovery at 4K.

Efficiency analysis. Table 6 compares inference speed at two resolutions. Our method is the fastest among the evaluated methods at 4K. At 1024×1024, it is faster than all evaluated generative baselines while remaining competitive with discriminative methods.

Table 2: **Depth accuracy on the high-resolution Synth4K benchmark,** evaluated over the full valid region (Overall) and high-frequency regions. Results for InfiniDepth† are reported with a 504×896 input and a 4K output, following its original paper. All other results are reported with 4K inputs and outputs. Our method demonstrates leading performance at high resolution.

	Method	Synth4K-1		Synth4K-2		Synth4K-3		Synth4K-4		Synth4K-5	
		AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑
Overall	DA3	38.3	45.7	29.7	51.2	32.2	53.2	23.7	63.9	23.8	58.9
	MoGe-2	22.1	67.1	16.6	76.6	18.0	79.3	14.1	82.9	11.0	85.9
	PPD	15.9	79.4	11.7	87.3	12.7	87.7	11.9	88.2	9.1	93.1
	PointDiT	44.2	42.6	40.4	40.2	47.7	36.6	29.6	55.6	39.1	38.4
	URGT	15.0	81.1	10.6	87.5	15.3	84.9	7.4	94.0	7.6	92.6
	InfiniDepth	31.3	59.0	21.7	69.6	33.0	56.7	17.5	80.8	29.9	65.3
	InfiniDepth†	**11.6**	**86.8**	9.4	88.2	12.2	85.2	**7.1**	**94.8**	**5.4**	95.6
	Ours (1024)	15.0	80.3	11.0	87.5	11.3	87.8	9.5	91.4	8.9	92.2
	Ours (MR)	12.3	85.3	**8.4**	**92.4**	**8.5**	**92.9**	7.7	93.6	6.7	**95.7**
High-frequency	DA3	45.6	41.8	36.4	45.2	44.8	44.1	27.9	58.0	27.8	56.6
	MoGe-2	29.4	58.0	24.7	62.7	31.5	61.9	19.9	72.1	15.5	78.0
	PPD	29.4	62.3	24.5	65.0	31.1	65.0	20.3	73.5	18.4	75.6
	PointDiT	49.0	42.1	38.8	42.9	53.0	39.0	34.1	49.6	32.7	47.5
	URGT	30.0	62.3	28.3	63.1	44.6	61.7	20.4	76.9	19.6	74.4
	InfiniDepth	51.1	44.2	36.9	49.6	56.4	39.3	36.2	64.4	49.2	46.1
	InfiniDepth†	32.1	67.3	28.2	64.7	34.9	63.5	28.2	76.7	19.3	78.8
	Ours (1024)	27.4	62.6	24.3	64.7	28.7	63.6	19.4	74.7	17.2	74.7
	Ours (MR)	**22.8**	**69.0**	**21.3**	**71.0**	**24.0**	**70.7**	**16.7**	**78.8**	**13.5**	**81.8**

Table 3: **Boundary F1↑ (radius = 1) results on Synth4K.** Our multi-resolution model achieves the highest scores across all five subsets, demonstrating excellent depth boundary quality.

Method	Synth4K-1	Synth4K-2	Synth4K-3	Synth4K-4	Synth4K-5
DA3	8.7	10.8	9.2	11.1	14.1
MoGe-2	15.7	13.1	13.2	14.9	18.2
PPD	15.0	18.4	15.8	18.7	16.1
PointDiT	20.7	21.8	17.4	22.1	20.1
URGT	20.9	16.0	15.0	18.2	21.4
InfiniDepth	15.7	20.0	13.1	23.6	21.7
InfiniDepth†	11.2	7.1	8.3	13.3	11.0
Ours (1024)	22.5	22.2	18.5	22.5	24.9
Ours (MR)	**31.1**	**29.5**	**25.7**	**30.4**	**32.5**

4.3 Ablation Studies

Decoding across different latent priors. To evaluate the effectiveness of pixel diffusion decoding across different latent priors, we compare Lotus-2 and two Flux.2-depth variants trained with latent- and pixel-space supervision, respectively. For each prior, we evaluate its standalone predictions and predictions obtained with an adapted PiD decoder. As shown in Table 4, PiD reduces overall AbsRel across all four datasets and improves both HF metrics on Synth4K-1/2 for all three priors, demonstrating its effectiveness in recovering fine-grained depth across different latent representations. Among these priors, pixel-supervised Flux.2-depth provides the lowest overall AbsRel on all four datasets and the best HF metrics on both Synth4K subsets after pixel decoding.

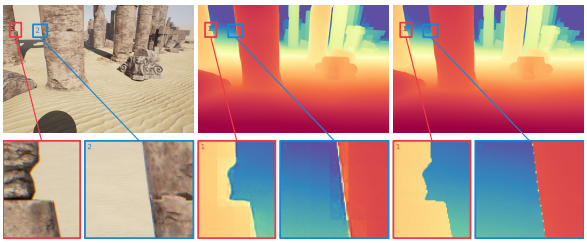
Effectiveness of continuity-promoting module. We compare the pixel decoder with and without the continuity-promoting module to assess its effect on patch-induced artifacts. As shown in Figure 4, the variant without the module exhibits discontinuities at patch-size intervals, especially near abrupt depth transitions. The module suppresses these artifacts through spatial feature mixing.

Table 4: Pixel diffusion decoding across different latent priors. Flux.2-Depth denotes our depth-adapted latent diffusion model. AbsRel^H and δ_1^H denote AbsRel and δ_1 evaluated in high-frequency masked regions, respectively.

Configuration	ETH3D		DIODE		Synth4K-1				Synth4K-2			
	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel^H ↓	δ_1^H ↑	AbsRel↓	δ_1 ↑	AbsRel^H ↓	δ_1^H ↑
Lotus-2	5.1	97.5	6.1	95.2	28.0	62.7	47.0	45.1	24.7	68.0	46.1	40.5
+ Pixel diffusion	4.1	98.3	4.8	96.9	17.9	74.8	31.3	57.4	13.2	83.7	25.8	60.1
Flux.2-Depth (latent loss)	4.5	98.8	5.6	96.4	22.7	65.1	31.4	53.8	17.7	74.4	31.6	51.9
+ Pixel diffusion	3.3	99.1	4.2	**98.0**	15.9	78.8	29.4	59.8	12.6	84.9	26.6	60.9
Flux.2-Depth (pixel loss)	3.6	**99.2**	4.6	97.7	19.9	70.7	29.1	56.9	16.6	77.8	29.4	52.4
+ Pixel diffusion	**3.1**	**99.2**	**3.9**	97.9	**15.0**	**80.3**	**27.4**	**62.6**	**11.0**	**87.5**	**24.3**	**64.7**

Table 5: **Effect of patch size in pixel diffusion.** We use a patch size of 16 for the best accuracy. Despite reduced boundary detail (F1), a patch size of 32 retains competitive overall depth accuracy while providing a 3.1× speedup at 3840×2112 resolution.

Patch	ETH3D		DIODE		Synth4K-1					Efficiency: Time (ms)↓	
	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	F1↑	AbsRel^H ↓	δ_1^H ↑	1024^2	3840×2112
16	3.1	99.1	3.9	97.9	12.3	85.3	31.1	22.8	69.0	204	1,439
32	3.3	98.9	4.0	97.8	12.7	85.1	17.6	25.4	66.8	172	468
64	4.1	98.2	4.6	97.2	13.1	84.4	9.2	28.1	63.7	170	362



RGB w/o module w/ module

Figure 4: **Ablation study of the proposed continuity-promoting module.** The module effectively mitigates discontinuity artifacts between patches.

Table 6: **Runtime comparison (ms↓)** at 1024×1024 and 3840×2160.

	Method	1024×1024	3840×2160
Discriminative	DA3	105	1,784
	MoGe-2	82	1,637
	URGT	-	24,306
	InfiniDepth	142	1,740
Generative	Marigold	614	13,769
	Lotus-2	7,949	106,314
	PPD	899	27,577
	PointDiT	221	5,095
	Ours	**204**	**1,477**

Patch size in pixel diffusion. Larger patches reduce the number of tokens and accelerate inference, but compromise depth accuracy. We study this trade-off with patch sizes of 16, 32, and 64. As shown in Table 5, increasing the patch size degrades prediction quality, with boundary and HF metrics showing a greater loss of detail than the overall depth metrics suggest. Notably, P32 retains similar overall accuracy to P16 while providing a $3.1\times$ speedup at 3840×2112 resolution, with degradation concentrated in fine-detail quality. While we adopt P16 to prioritize fine-grained geometry, P32 offers an efficient alternative for applications with less requirements for depth details.

5 Conclusion

We presented EagleDepth, a framework that combines depth-adapted latent diffusion with pixel diffusion for efficient fine-grained depth estimation. The latent branch predicts coarse depth latents from downsampled RGB inputs, while the pixel branch generates depth directly at the target resolution conditioned on these latents. This design retains learned geometric priors while reducing detail loss from latent-space modeling and VAE reconstruction. It also avoids the substantial overhead of VAE decoding at high resolutions. We train two variants: one at approximately 1024^2 pixels and a mixed-resolution extension supporting prediction up to 4K (2160×3840). Experiments on five commonly used depth estimation datasets and Synth4K demonstrate strong overall accuracy and better preservation of fine structures and object boundaries.

AI USE STATEMENT

We used generative AI tools to polish the language and improve the clarity of technical descriptions in this manuscript. The authors reviewed all AI-assisted revisions, checking that the wording preserved the intended technical meaning and remained consistent with the methods and reported results. The authors take full responsibility for the final manuscript, including all AI-assisted text and the claims it conveys.

REPRODUCIBILITY STATEMENT

The main text and appendix provide the details necessary to reproduce our results, including the model architecture, training procedure, data processing, and evaluation protocols. We will publicly release all source code and model weights to facilitate independent reproduction and further research.

REFERENCES

Black Forest Labs. FLUX.2 [klein]: Towards interactive visual intelligence. https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence, 2026.

Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations*, volume 2025, pp. 75602–75637, 2025.

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

Wenqing Cui, Zhenyu Li, Mykola Lavreniuk, Jian Shi, Ramzi Idoughi, Xiangjun Tang, and Peter Wonka. Any resolution any geometry: From multi-view to multi-patch. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12576–12585, 2026.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2432–2443. IEEE, 2017.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, volume 2025, pp. 34668–34684, 2025.

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 *European Conference on Computer Vision*, pp. 241–258. Springer, 2024.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361. IEEE, 2012.

Samiran Gode, Abhijeet Nayak, Débora NP Oliveira, Michael Krawez, Cordelia Schmid, and Wolfram Burgard. Flownav: Combining flow matching and depth priors for efficient navigation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 17762–17768. IEEE, 2025.

Jose L Gómez, Manuel Silva, Antonio Seoane, Agnes 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.

Jing He, Haodong Li, Mingzhi Sheng, and Ying-Cong Chen. Lotus-2: Advancing geometric dense prediction with powerful image generative model. *arXiv preprint arXiv:2512.01030*, 2025a.

Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Zhang, Bingbing Liu, and YingCong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. In *International Conference on Learning Representations*, volume 2025, pp. 89454–89467, 2025b.

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. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2005–2015. IEEE, 2025.

Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9492–9502. IEEE, 2024a.

Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9492–9502. IEEE, 2024b.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

Hoang-An Le, Thomas Mensink, Partha Das, Sezer Karaoglu, and Theo Gevers. Eden: Multimodal synthetic dataset of enclosed garden scenes. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1578–1588. IEEE, 2021.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026.

Zhenyu Li, Shariq Farooq Bhat, and Peter Wonka. Patchfusion: An end-to-end tile-based framework for high-resolution monocular metric depth estimation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10016–10025. IEEE, 2024.

Zhenyu Li, Wenqing Cui, Shariq Bhat, and Peter Wonka. Patchrefiner v2: Fast and lightweight real-domain high-resolution metric depth estimation. In *International Conference on Learning Representations*, volume 2026, pp. 117738–117753, 2026.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

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

Yifan Lu, Qi Wu, Jay Zhangjie Wu, Zian Wang, Huan Ling, Sanja Fidler, and Xuanchi Ren. Pid: Fast and high-resolution latent decoding with pixel diffusion. *arXiv preprint arXiv:2605.23902*, 2026.

Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4981–4991. IEEE, 2023.

Wenzhe Ouyang, Xiaolin Song, Bailan Feng, and Zenglin Xu. Octocc: High-resolution 3d occupancy prediction with octree. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4369–4377, 2024.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12159–12168. IEEE, 2021.

Adrian Redder, Huy Xuan Pham, Andriy Sarabakha, Erdal Kayacan, et al. Vds-nav: Volumetric depth-based safe navigation for aerial robots–bridging the sim-to-real gap. *IEEE Robotics and Automation Letters*, 2025.

Zeyu Ren, Zeyu Zhang, Wukai Li, and Hao Tang. Anydepth: Depth estimation made easy. 2026.

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 *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10892–10902. IEEE, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3260–3269, 2017.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pp. 746–760. Springer, 2012.

Bin Tan, Changjiang Sun, Xiage Qin, Hanat Adai, Zelin Fu, Tianxiang Zhou, Han Zhang, Yinghao Xu, Xing Zhu, Yujun Shen, and Nan Xue. Masked depth modeling for spatial perception. *arXiv preprint arXiv:2601.17895*, 2026.

Shitao Tang, Jiacheng Chen, Dilin Wang, Chengzhou Tang, Fuyang Zhang, Yuchen Fan, Vikas Chandra, Yasutaka Furukawa, and Rakesh Ranjan. Mvdiffusion++: A dense high-resolution multi-view diffusion model for single or sparse-view 3d object reconstruction. In *European Conference on Computer Vision*, pp. 175–191. Springer, 2024.

Rawmantic Team. Rawmantic 3d training field: From the real world, for the real world. *Rawmantic*, 2026.

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.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-Ω. *arXiv preprint arXiv:2605.15195*, 2026a.

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 *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2021.

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. *Advances in Neural Information Processing Systems*, 38:35928–35959, 2026b.

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 *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4909–4916. IEEE, 2020.

Gangwei Xu, Haotong Lin, Hongcheng Luo, Xianqi Wang, Jingfeng Yao, Lianghui Zhu, Yuechuan Pu, Cheng Chi_, Haiyang Sun, Bing Wang, et al. Pixel-perfect depth with semantics-prompted diffusion transformers. *Advances in Neural Information Processing Systems*, 38:174731–174755, 2026a.

Haofei Xu, Rundi Wu, Philipp Henzler, Nikolai Kalischek, Michael Oechsle, Fabian Manhardt, Marc Pollefeys, Andreas Geiger, Federico Tombari, and Michael Niemeyer. Pointdit: Pixel-space diffusion for monocular geometry estimation. *arXiv preprint arXiv:2607.02515*, 2026b.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in neural information processing systems*, 37:21875–21911, 2024.

Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5683–5692. IEEE, 2019.

Hao Yu, Haotong Lin, Jiawei Wang, Jiaxin Li, Yida Wang, Xueyang Zhang, Yue Wang, Xiaowei Zhou, Ruizhen Hu, and Sida Peng. Infinidepth: Arbitrary-resolution and fine-grained depth estimation with neural implicit fields. *arXiv preprint arXiv:2601.03252*, 2026a.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14273–14282, 2026b.

Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. Depthvla: Enhancing vision-language-action models with depth-aware spatial reasoning. *arXiv preprint arXiv:2510.13375*, 2025.

Hongfei Zhang, Harold Haodong Chen, Chenfei Liao, Jing He, Zixin Zhang, Haodong Li, Yihao Liang, Kanghao Chen, Bin Ren, Xu Zheng, et al. Dvd: Deterministic video depth estimation with generative priors. In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*, 2026a.

Tianbao Zhang, Zhenyu Liang, Zhenbo Song, Nana Wang, Xiaomei Zhang, Xudong Cai, Zheng Zhu, Kejian Wu, Gang Wang, and Zhaoxin Fan. 2k retrofit: Entropy-guided efficient sparse refinement for high-resolution 3d geometry prediction. In *European Conference on Computer Vision*, pp. 150–169. Springer, 2026b.

Tianbao Zhang, Zeyu Liu, Shuyu Wu, Fanxing Li, Zhaoxin Fan, Danping Zou, et al. Litemvs: Efficient multi-view stereo with foundation distillation and expert aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4520–4529, 2026c.

Yi Zhang, Weichao Qiu, Qi Chen, Xiaolin Hu, and Alan Yuille. Unrealstereo: Controlling hazardous factors to analyze stereo vision. In *2018 International Conference on 3D Vision (3DV)*, pp. 228–237. IEEE, 2018.

Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.

A Architecture Details

A.1 Latent backbone: FLUX.2

We use FLUX.2-klein-4B (Labs, 2025) as the latent backbone. This pretrained rectified-flow transformer contains approximately four billion parameters, with five double-stream blocks followed by twenty single-stream blocks. All blocks use a hidden dimension of 3072 and 24 attention heads. The double-stream blocks maintain separate image and text projections and feed-forward networks, while jointly attending to tokens from both modalities. The single-stream blocks process the concatenated sequence of image and text tokens with parallel attention and feed-forward branches. The backbone uses query and key normalization, rotary positional embeddings, and timestep-conditioned feature modulation. Its Qwen3-4B text encoder concatenates features from three intermediate layers into 7680-dimensional text tokens, which are projected to the transformer width.

The accompanying FLUX.2 autoencoder uses a convolutional encoder–decoder with residual blocks and a 32-channel latent representation at one-eighth of the image height and width. Packing each 2×2 latent neighborhood into the channel dimension produces 128-channel features at a spatial stride of 16. The encoder uses the posterior mean and normalizes the packed features using fixed pretrained channel statistics. For an input image of size $H \times W$, the resulting latent representation has spatial dimensions $H/16 \times W/16$ and 128 channels. The transformer predicts a velocity in this same representation.

We initialize the latent predictor from the released distilled checkpoint and adapt its transformer blocks with rank-256 LoRA. The encoded RGB image serves as the input state at $t^\star = 0.5$, and the depth latent is obtained by the single flow update in Eq. 3. For text conditioning, we use the fixed prompt "A dense grayscale monocular depth map of the scene, far surfaces brighter and near surfaces darker." Its embedding is computed once and cached. The autoencoder remains frozen; its decoder is used for the coarse-depth supervision described in the main paper. During pixel adaptation and final inference, the latent predictor remains frozen and operates at approximately 512^2 input pixels, passing its predicted depth latent directly to the pixel diffusion decoder.

A.2 Pixel backbone: PiD

We use pretrained PiD (Lu et al., 2026) as the pixel backbone, adopting its FLUX.2-compatible v1.5 architecture. Built on PixelDiT (Yu et al., 2026b), PiD combines a patch transformer with a pixel pathway to jointly perform latent decoding and spatial upsampling. The patch transformer processes 16×16 image patches using 14 MM-DiT blocks with hidden dimension 1536 and 24 attention heads, allowing image and text tokens to interact through joint attention. The pixel pathway uses two PiT blocks with 16-dimensional pixel features, conditioned on the patch representations through adaptive normalization. Each PiT block compresses the features within each patch for attention across patches, then expands them back for pixel updates.

A convolutional adapter maps the conditioning latent to features aligned with the image patch grid. For FLUX.2 latents, the packed 128-channel representation is first unpacked into 32 channels and resized by nearest-neighbor interpolation to $H/p \times W/p$, where $p = 16$. Two 3×3 convolutions with an intermediate SiLU activation and four residual blocks extract features with 1024 channels. These features are flattened into tokens and independently projected to the backbone width:

$$\begin{aligned} \boldsymbol{z}_{\mathrm{grid}} &= \mathrm{Resize}\big(\mathrm{Unpack}_{2\times2}(\tilde{\boldsymbol{z}}_{\sigma_{\mathrm{lat}}})\big)\,, \\ \boldsymbol{\ell}^{(k)} &= \mathrm{Linear}_k(\mathrm{Flatten}(\mathrm{ConvBlocks}(\boldsymbol{z}_{\mathrm{grid}})))\,, \qquad k = 1, \ldots, 7, \end{aligned} \tag{7}$$

where $\tilde{\boldsymbol{z}}_{\sigma_{\mathrm{lat}}}$ denotes the conditioning latent at noise level σ_{lat}, and $\mathrm{ConvBlocks}$ denotes the convolutional feature extractor described above. Each $\boldsymbol{\ell}^{(k)} \in \mathbb{R}^{N \times 1536}$ contains one conditioning feature per image patch, with $N = HW/p^2$.

Original PiD trains with corrupted conditions $\tilde{\boldsymbol{z}}_{\sigma_{\mathrm{lat}}} = (1-\sigma_{\mathrm{lat}})\boldsymbol{z} + \sigma_{\mathrm{lat}}\boldsymbol{\epsilon}$, where $\boldsymbol{z}$ is the uncorrupted latent and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, to support decoding from partially denoised latents. A sigma-aware gate regulates latent injection before alternating MM-DiT blocks:

$$\begin{aligned} \boldsymbol{g}^{(k)} &= \mathrm{sigmoid}\Big(\mathrm{Linear}_k^{\mathrm{gate}}\Big([\boldsymbol{s}^{(k)}; \boldsymbol{\ell}^{(k)}]\Big) - \alpha_k \sigma_{\mathrm{lat}}\Big)\,, \\ \tilde{\boldsymbol{s}}^{(k)} &= \boldsymbol{s}^{(k)} + \boldsymbol{g}^{(k)} \odot \boldsymbol{\ell}^{(k)}, \end{aligned} \tag{8}$$

Table 7: **Training datasets and sample selection strategies.** We report depth resolutions ($H \times W$), selection criteria, and the number of selected RGB–depth pairs for each dataset.

Dataset	Depth resolution	Selection	Samples
Hypersim	768×1024	AnyDepth filtering	26,695
VKITTI2	375×1242	AnyDepth filtering	12,643
TartanAir	480×640	AnyDepth filtering	186,693
IRS	540×960	AnyDepth filtering	68,211
EDEN	480×640	Official subset	12,520
RobbyReal	480×640, 720×1280	Every tenth frame	139,575
Rawmantic3D	1080×1920	Every tenth frame	68,686
UnrealStereo4K	2160×3840	All samples	8,000
UrbanSyn	1024×2048	All samples	7,539
Spring	2160×3840	All samples	5,000
Total			**535,562**

where $\boldsymbol{s}^{(k)}$ denotes the image tokens before block $2k-1$, $[\cdot;\cdot]$ denotes channel concatenation, and $\alpha_k > 0$ is learned. In the v1.5 implementation, each gate produces one scalar per token, broadcast across channels. The content term learns the injection strength, while the noise-dependent offset reduces it as σ_{lat} increases. An additional projection and gate condition the patch features supplied to both PiT blocks.

For our task, we use the observed RGB image as the pixel input state and the coarse predictor's depth latent $\hat{\boldsymbol{z}}^{L}_{\mathrm{depth}}$ as the conditioning signal. For simplicity, we treat this predicted latent as clean: no noise is added, and $\sigma_{\mathrm{lat}} = 0$ throughout training and inference. Thus, we retain content-dependent gating without noise-level modulation. We freeze the coarse predictor and fine-tune PiD and its adapter for single-step depth prediction at $t^{\star} = 0.5$, retaining a fixed text embedding cache. The decoder directly predicts the clean sample (x-pred), with the continuity-promoting module inserted before the original output projection as described in the main paper. Its three output channels are averaged to obtain the depth map.

B Dataset Details

To improve training efficiency and reduce storage requirements, we filter and subsample the larger training datasets. The aim is to retain informative samples while reducing low-quality supervision and redundant observations.

For Hypersim, VKITTI2, TartanAir, and IRS, we adopt training subsets based on the quality filtering of AnyDepth (Ren et al., 2026), which screens samples according to valid-depth coverage, depth-value distribution, and gradient continuity. For EDEN, we use the smaller sample subset provided by the dataset authors. For RobbyReal and Rawmantic3D, we observe substantial similarity between neighboring frames within a scene and therefore retain every tenth frame from each sequence. For UnrealStereo4K, UrbanSyn, and Spring, we retain all available training samples because these datasets are relatively small and provide valuable supervision for high-resolution depth prediction. Table 7 summarizes the number of samples used from each dataset, totaling 535,562 image and depth pairs (approximately 536K).

C Training Details

C.1 Data Augmentation

We use only random horizontal flipping and gamma augmentation to improve robustness to scene orientation and photometric variations. Both are applied with probability 0.5, with gamma augmentation defined as $I' = I^{\gamma}$ for RGB intensities $I \in [0,1]$ and $\gamma \sim \mathcal{U}(0.8, 1.2)$. The high-resolution stream uses horizontal flipping only.

Table 8: **Training resolution buckets for our models.** Aspect ratios are given as $W : H$ and image dimensions as $H \times W$. All dimensions are divisible by 16 to accommodate the models' patch size.

	Latent branch	Pixel decoder			
Aspect ratio	512-scale	1024-scale	High-res. low	High-res. middle	High-res. high
$1:1$	512×512	1024×1024	1536×1536	2048×2048	2160×2160
$4:3$	432×576	864×1152	1344×1792	1776×2368	2160×2880
$3:2$	416×624	832×1248	1248×1872	1664×2496	2144×3216
$16:9$	384×688	720×1280	1152×2048	1584×2816	2160×3840
$2:1$	368×736	720×1440	1088×2176	1456×2912	1920×3840
$10:3$	288×960	576×1920	864×2880	1104×3680	1152×3840

Table 9: Component-wise runtime (ms↓) and peak GPU memory (GB↓) analysis. All runtime results are evaluated on an NVIDIA RTX 4090 GPU. CPM denotes the continuity-promoting module.

Component	512×512		1024×1024		2048×2048		2160×3840	
	Time	Memory	Time	Memory	Time	Memory	Time	Memory
VAE Encoder	16	11.9	83	13.2	424	18.0	1058	24.4
DiT	102	11.8	349	12.4	1733	14.7	4423	17.6
VAE Decoder	34	12.1	159	13.8	732	20.7	1668	29.8
Pixel Diffusion	37	11.8	78	12.4	487	14.8	1298	18.0
CPM	2	11.7	7	11.9	39	13.1	77	14.7

C.2 Multi-Aspect-Ratio and Mixed-Resolution Training

Aspect-ratio bucketing. Excessive stretching or compression distorts object shapes and proportions, potentially weakening semantic and geometric cues for depth estimation. At each resolution level, we therefore use six representative aspect ratios: $1:1$, $4:3$, $3:2$, $16:9$, $2:1$, and $10:3$, expressed as width to height. Images are cropped to the selected aspect ratio before resizing to preserve object proportions. During bucketed inference, images whose height exceeds their width are rotated by 90°, and the predicted depth is rotated back to the original orientation. Table 8 lists the corresponding image dimensions.

Mixed-resolution sampling. Coarse depth prediction operates at approximately 512^2 pixels to provide scene-level depth structure. The pixel diffusion decoder is first trained at approximately 1024^2 pixels, allowing it to learn depth prediction at moderate computational cost. We then continue training with mixed resolutions up to 4K (2160×3840) to adapt the model to a wider range of output resolutions, while keeping the coarse predictor frozen at the 512-scale setting. The high-resolution buckets comprise three levels (low, medium, and high) which are trained jointly with 1024-scale replay to mitigate forgetting. The high-resolution stream uses only UnrealStereo4K, UrbanSyn, Spring, and Rawmantic3D to learn fine geometric details and depth boundaries. The four levels (1024-scale and the three high-resolution levels) are sampled with equal probability, each accounting for 25% of training batches in expectation.

D Additional Results

D.1 More Qualitative Results

Figure 5 presents point-cloud reconstructions from our predicted depth maps, obtained using the same depth-to-point-cloud conversion procedure as PPD (Xu et al., 2026a). Figures 6, 7, and 8 provide additional qualitative comparisons with other methods, further illustrating our method's ability to recover fine geometric details while preserving overall scene structure.

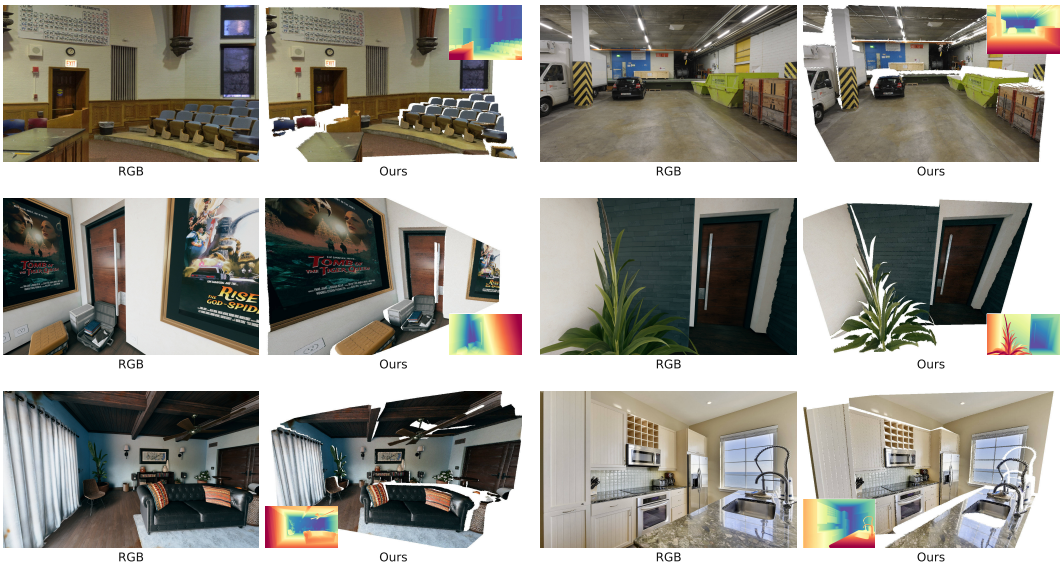


Figure 5: **Point cloud results of our method.**

D.2 Component-wise efficiency.

Table 9 provides a component-wise analysis of the VAE encoder, FLUX.2 DiT, VAE decoder, PiD, and continuity-promoting module (CPM). Beyond its advantages in depth quality, the pixel diffusion decoder reduces decoding cost compared with the VAE decoder, including at high output resolutions. The continuity-promoting module further suppresses patch-boundary artifacts with modest computational overhead.

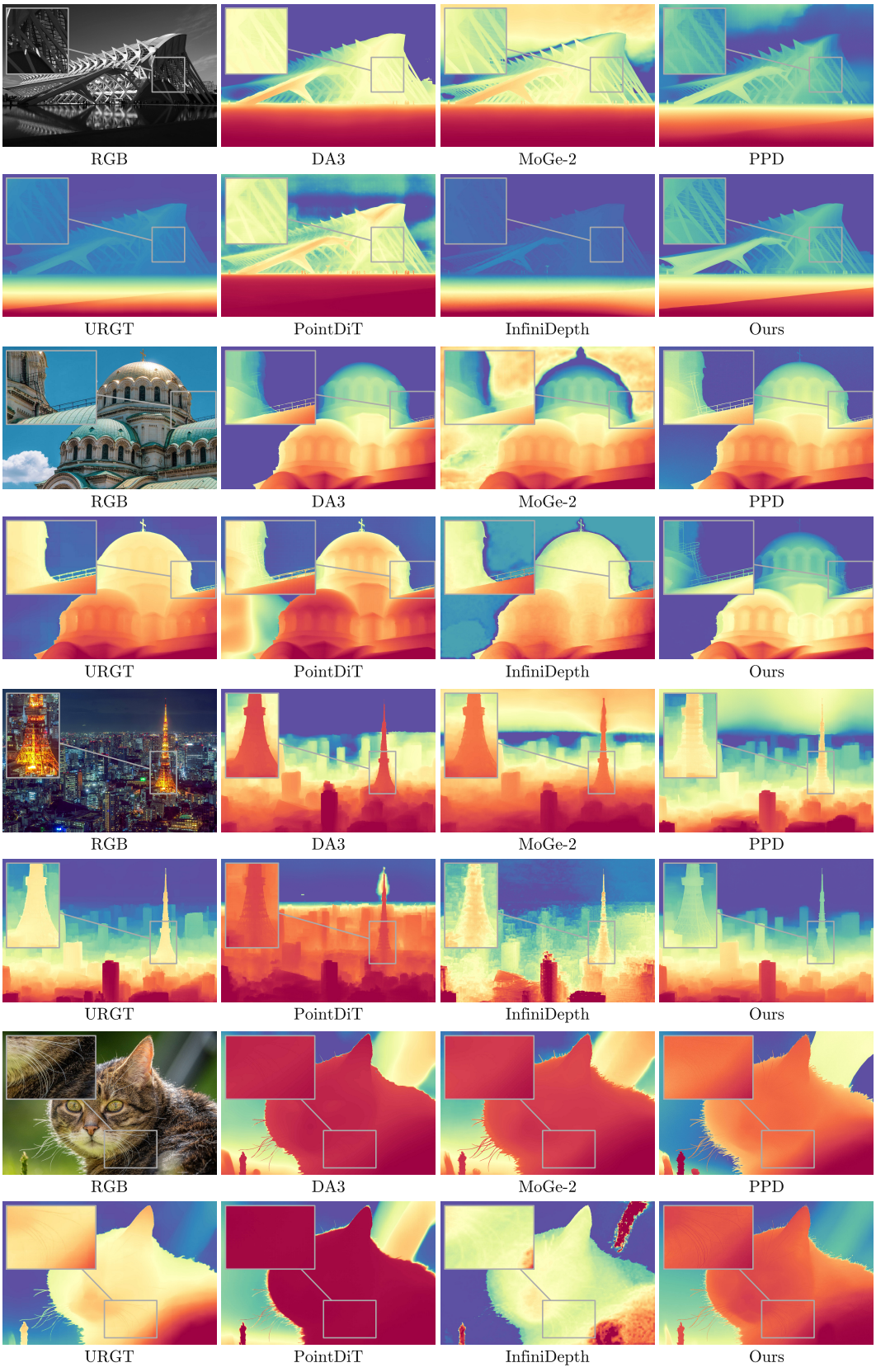


Figure 6: **More visual results (1).** Zoom in for better visibility.

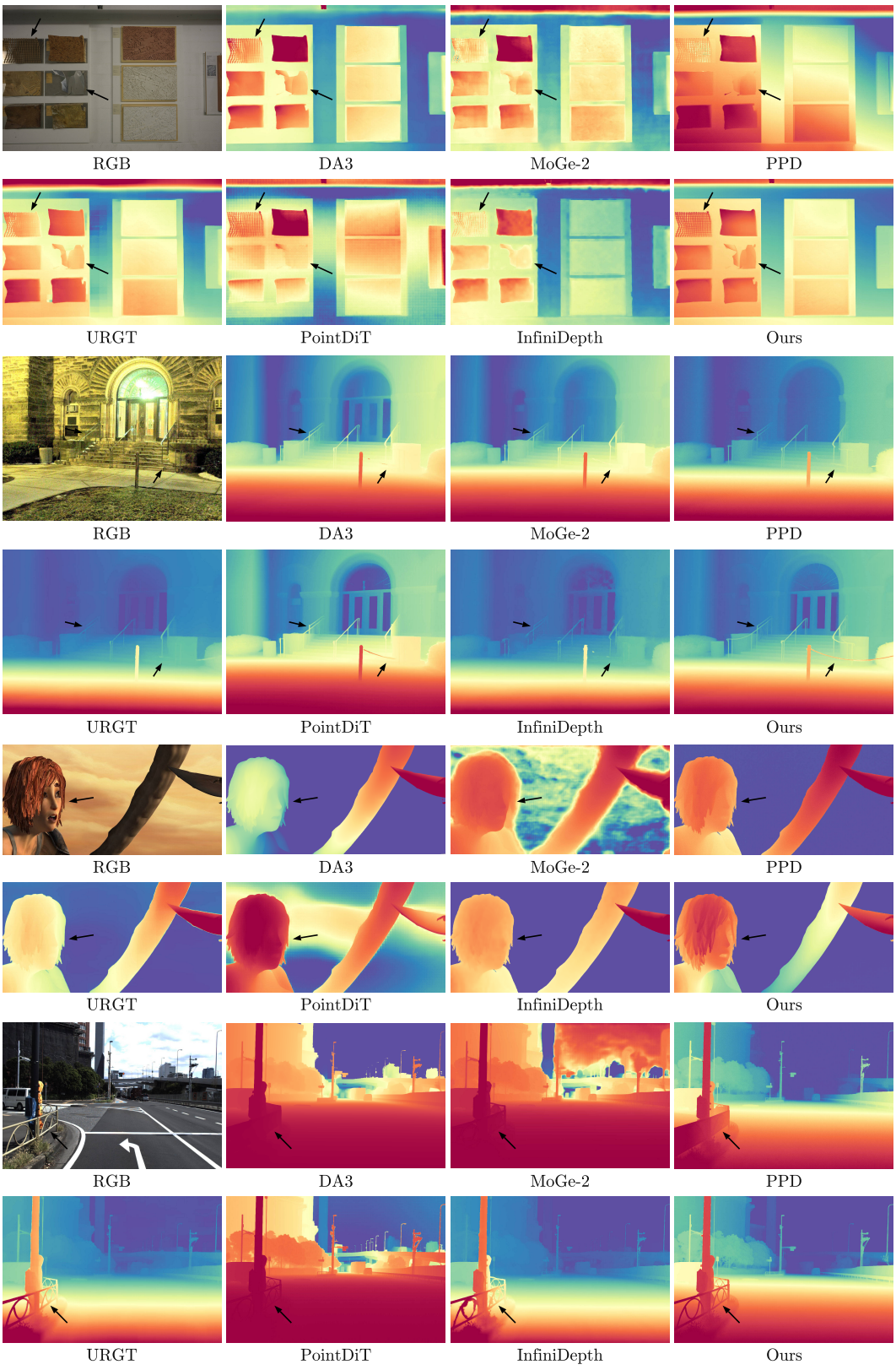


Figure 7: **More visual results (2).** Zoom in for better visibility.

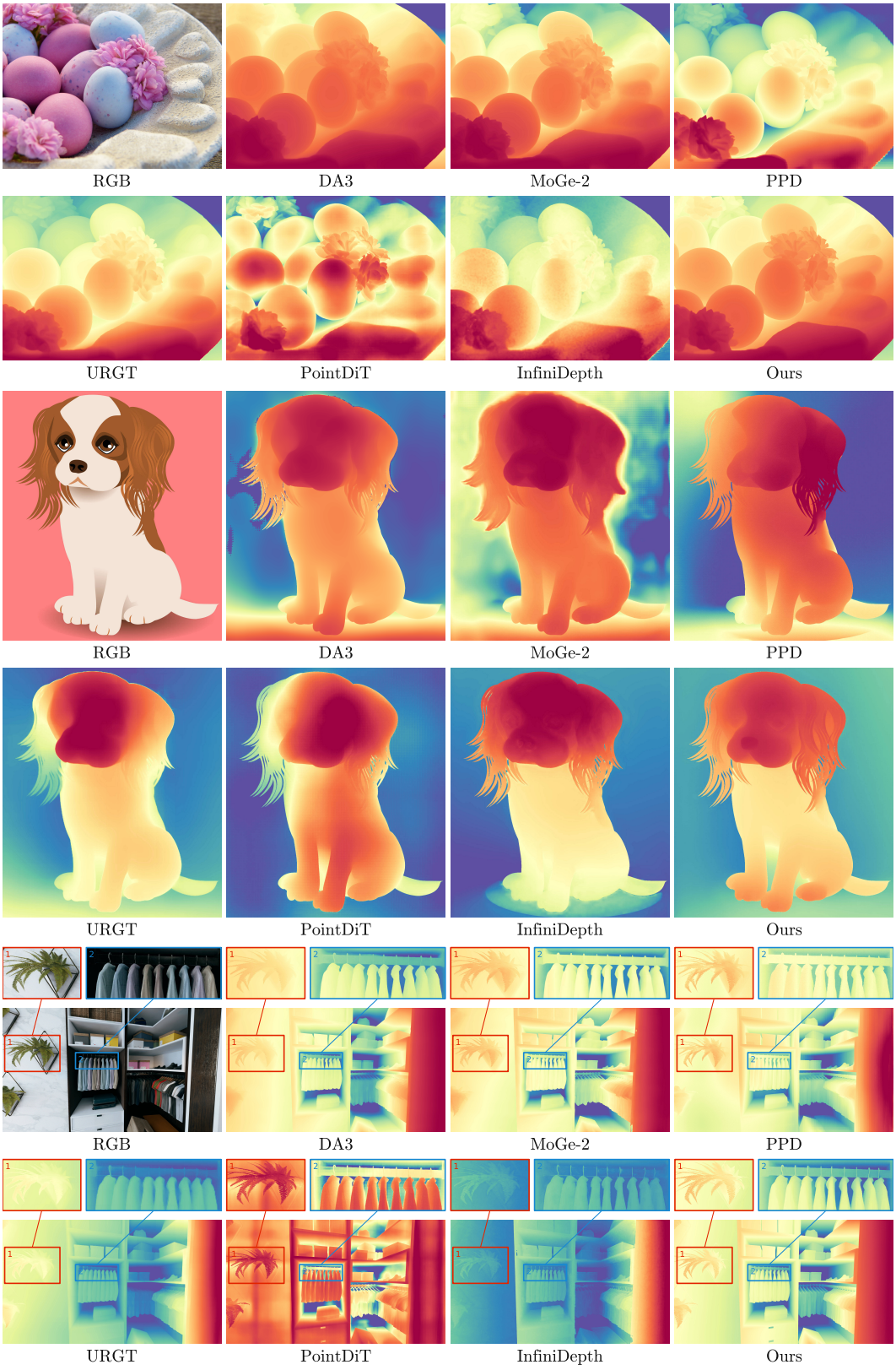


Figure 8: **More visual results (3).** Zoom in for better visibility.