

# Any Resolution Any Geometry: From Multi-View To Multi-Patch

## Supplementary Material

In this supplementary material, we provide additional implementation and dataset details (including training hyper-parameters, evaluation protocols, and metric definitions), extended ablation studies on data augmentation and GridMix sampling, more qualitative visualizations at multiple resolutions (2K–8K) and out-of-domain images, and a discussion of current limitations and directions for future work. We also refer readers to our website (<https://dreamaker-mrc.github.io/Any-Resolution-Any-Geometry>), which provides an interactive comparison of our predictions.

### A. Implementation Details

#### A.1. Training Details

We adopt Depth-Anything V2 (DA2) [29] and Metric3D V2 [8] as the coarse prediction backbones. Training is conducted on 7,592 samples from the UnrealStereo4K [24] dataset for 80K iterations using 4 NVIDIA A100 GPUs with a batch size of 1. We use the AdamW optimizer with a learning rate of  $1 \times 10^{-5}$ , weight decay of  $1 \times 10^{-6}$ , and apply gradient norm clipping with a maximum  $\ell_2$  norm of 35 to stabilize training; gradient checkpointing is enabled to reduce memory consumption.

During training, we operate on fixed-size patches with a resolution of  $540 \times 960$ . Our multi-task objective combines a depth loss  $\mathcal{L}_{\text{depth}}$  with a normal loss  $\mathcal{L}_{\text{normal}}$ , and the final loss is given by

$$\mathcal{L}_{\text{total}} = \lambda_{\text{depth}} \mathcal{L}_{\text{depth}}(D^{\text{refined}}, D^{\text{gt}}) + \lambda_{\text{normal}} \mathcal{L}_{\text{normal}}(\mathbf{n}^{\text{refined}}, \mathbf{n}^{\text{pseudo}}), \quad (10)$$

where we set  $\lambda_{\text{depth}} = 1$ ,  $\lambda_{\text{normal}} = 0.01$  to balance the contribution of the normal supervision.

#### A.2. Datasets

**UnrealStereo4K.** The UnrealStereo4K dataset [24] provides stereo image pairs at 4K resolution ( $2160 \times 3840$ ), each with dense and boundary-preserving ground-truth annotations. All scenes are rendered in Unreal Engine using the UnrealCV plugin across eight virtual environments, offering diverse geometry, materials, and lighting conditions. Following [11, 12], we first remove mislabeled samples using the Structural Similarity Index (SSIM) [28]. Following PRO [9], we employ the same Bias-Free Mask during training to ensure a fair comparison. The final depth ground truth is then computed from the provided disparity maps using the calibrated camera baseline and focal length.

**Middlebury.** The Middlebury 2014 dataset [21] provides high-resolution indoor scenes with accurate ground-truth

disparity and depth annotations. Following common practice, we select 23 stereo pairs with valid ground truth and convert the provided disparity maps into depth maps using the calibrated camera parameters.

**Booster.** The Booster dataset [19] contains high-resolution indoor images ( $3008 \times 4112$ ) featuring challenging specular, reflective, and transparent surfaces. We use the whole training set with GT for evaluation, resulting in 228 images.

**ETH3D.** The ETH3D high-resolution dataset [22] includes both indoor and outdoor scenes ( $6048 \times 4032$ ) with accurate ground-truth depth maps captured using LiDAR sensors.

#### A.3. Evaluation Details

**PatchRefiner.** We retrained PatchRefiner [12] on top of DA2 for a fair comparison. We evaluate under two configurations: (i)  $p = 16$ , which uses the same number of patches as our method, and (ii)  $p = 49$ , where additional patches are employed for test-time ensembling. Here,  $p$  denotes the number of patches used to reassemble the final depth map during inference.

**Consistency Error (CE).** When computing the consistency error (CE) between neighboring patches, we evaluate the discrepancy only within a 270-pixel-wide overlapping region at the patch boundaries.

**Pseudo Depth Boundary Error (PDBE).** Following SharpDepth [16], we evaluate depth boundary quality using the Pseudo Depth Boundary Error, decomposed into an accuracy term  $\epsilon_{\text{PDBE}}^{\text{acc}}$  and a completeness term  $\epsilon_{\text{PDBE}}^{\text{compl}}$ . Given a predicted depth map  $D^{\text{refined}}$  and ground-truth depth  $D^{\text{gt}}$ , we first normalize each map to  $[0, 1]$  and apply a Canny edge detector to obtain depth edges  $E_{\text{depth}}^{\text{refined}}$  and  $E_{\text{depth}}^{\text{gt}}$ . In addition, we convert depth to disparity, normalize them, and run Canny again to extract disparity edges  $E_{\text{disp}}^{\text{refined}}$  and  $E_{\text{disp}}^{\text{gt}}$ . The final ground-truth and predicted edge maps are then defined as the union of depth and disparity edges, i.e.,  $E^{\text{refined}} = E_{\text{depth}}^{\text{refined}} \vee E_{\text{disp}}^{\text{refined}}$  and  $E^{\text{gt}} = E_{\text{depth}}^{\text{gt}} \vee E_{\text{disp}}^{\text{gt}}$ .

We compute Euclidean distance transforms  $T^{\text{refined}}$  and  $T^{\text{gt}}$  on the complements of  $E^{\text{refined}}$  and  $E^{\text{gt}}$ , respectively, truncated to a local neighborhood of 10 pixels. The PDBE accuracy  $\epsilon_{\text{PDBE}}^{\text{acc}}$  measures how close each predicted edge is to the nearest ground-truth edge:

$$\epsilon_{\text{PDBE}}^{\text{acc}} = \frac{\sum_x T^{\text{refined}}(x) E^{\text{gt}}(x)}{\sum_x E^{\text{gt}}(x)}, \quad (11)$$

Table 6. Ablation on GridMix Patch Sampling Strategy.

Configurations ( $p_1, p_2, p_3, p_4$ )	AbsRel ↓	$\delta_1$ ↑	RMSE ↓	CE ↓
Depth-Anything v2	0.0812	0.924	2.86	—
(1,0,0,0)	0.0500	0.963	2.01	0.0648
(0.5,0.5,0,0)	0.0473	0.966	1.79	0.0436
(0,1,0,0)	0.0405	0.973	1.72	0.0447
(0.5,0,0.5,0)	0.0365	0.975	1.58	0.0440
(0,0,1,0)	0.0350	0.976	1.57	0.0443
(0.4,0.3,0.3,0)	0.0343	0.977	1.52	0.0457
(0.1,0.2,0.3,0.4)	<b>0.0295</b>	<b>0.982</b>	<b>1.35</b>	<b>0.0418</b>
(0,0,0.5,0.5)	0.0311	0.981	1.38	0.0435
(0,0,0,1)	0.0321	0.980	1.42	0.0635

while the PDBe completeness  $\epsilon_{\text{PDBe}}^{\text{compl}}$  quantifies how well ground-truth edges are recovered by the prediction:

$$\epsilon_{\text{PDBe}}^{\text{compl}} = \frac{\sum_x T^{gt}(x) E^{\text{refined}}(x)}{\sum_x E^{\text{refined}}(x)}. \quad (12)$$

In both cases, lower values indicate sharper and better aligned depth boundaries.

## B. More Ablation Study

To empirically determine the optimal configuration for our proposed GridMix, we conduct a series of experiments focusing on different patch sampling probabilities ( $p_1, p_2, p_3, p_4$ ). As shown in Table 6, the configuration (0.1, 0.2, 0.3, 0.4) consistently outperforms other variants across all evaluated metrics. Specifically, it achieves the best performance not only in depth accuracy—indicated by AbsRel ↓,  $\delta_1$  ↑, and RMSE ↓—but also in the consistency metric CE ↓. These results indicate that probabilistic grid sampling effectively enhances inter-patch coherence while preserving fine geometric detail, leading to a more robust representation compared to the Depth-Anything v2 baseline.

## C. Extension to Any Resolution

Our patch-based formulation naturally supports arbitrary input resolutions at test time. Given a high-resolution image, we keep both the patch size and the transformer backbone fixed, and only scale the patch grid to cover the full image domain. This enables our model to handle 2K, 4K, 8K, and even higher resolutions without any resolution-specific retraining, while preserving local fine details and maintaining global geometric consistency across all patches.

As shown in Fig. 10, we apply the same model to in-the-wild images at 2K, 4K, and 8K resolutions. Across these settings, our approach consistently sharpens thin structures, refines object boundaries, and produces smoother, more coherent normal fields compared to the coarse predictions

from Depth-Anything V2 and Metric3D V2. In addition, Fig. 13 illustrates an 8K manga-style image that lies far outside the training domain. Even under this highly stylized, out-of-domain scenario, the model still recovers geometrically plausible depth and normals, highlighting the strong generalization ability of our multi-patch transformer for high-resolution single-image geometry estimation.

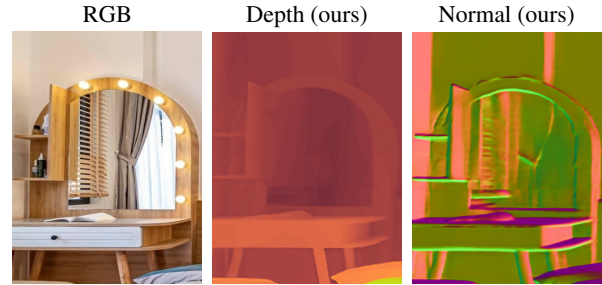


Figure 9. **Failure case on reflective surfaces.** Our model struggles to handle strongly reflective objects such as mirrors: the mirror region is incorrectly interpreted as a continuation of the surrounding geometry.

## D. More Qualitative Results

We further provide qualitative results on the Unreal-Stereo4K [24] dataset. Figure 14 shows three representative scenes, where each column corresponds to one sample and the rows visualize the RGB input, coarse depth from Depth-Anything V2 [29], depth refinements from PatchRefiner [12] and PRO [9], our refined depth, coarse normals from Metric3D V2 [8], and our refined normals. Across diverse scenes and lighting conditions, our method consistently sharpens depth, cleans up noisy regions, and recovers fine geometric structures such as thin objects, furniture details, and small decorations. On the normal estimation side, our model produces smoother and more coherent normal fields and better fine-details.

### D.1. Improvement over Base Models

While base models provide an accurate global scale, they are trained on low resolutions and lack the high-frequency details necessary to be good initial models for fine geometry. The marginal numerical gain stems from metrics like AbsRel/ $\delta_1$  being dominated by global scale and failing to reflect local precision. Furthermore, since real-world Ground Truth is highly sparse, edge-quality metrics cannot be applied to quantify our gains. Our method provides a critical geometric correction of over-smoothed boundaries (see Fig. 11) that global pixel-wise averages fail to capture. In general, qualitative comparisons better highlight the key gains of our method.



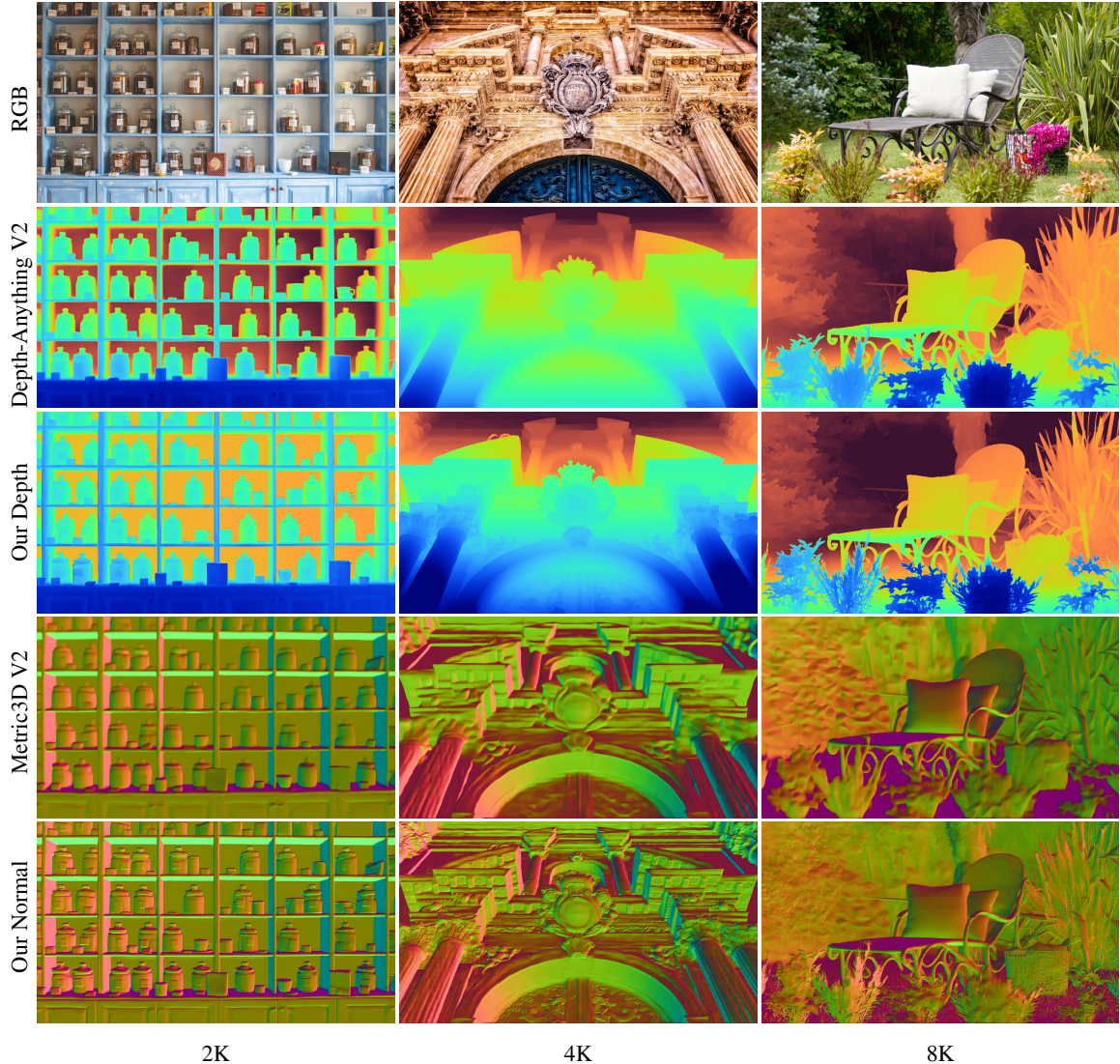


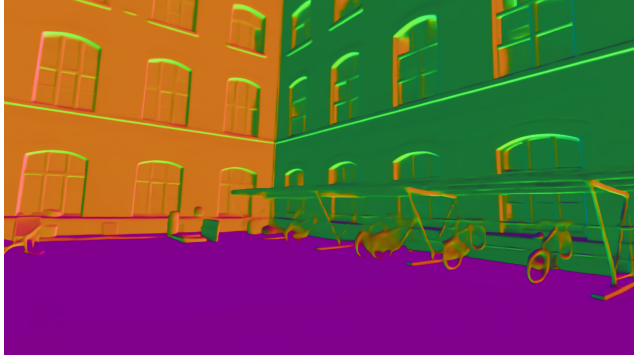
Figure 10. **Extension to arbitrary resolutions.** Each column shows a different input resolution (2K, 4K, 8K), and each row corresponds to the RGB input, coarse depth prediction by Depth-Anything V2, our refined depth prediction, coarse normal prediction by Metric3D V2, and our refined normal prediction, respectively.

## D.2. Applications: Lens Blur

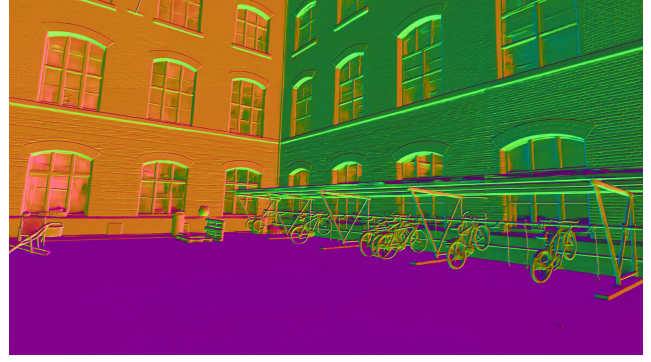
We simulate a lens blur effect using depth maps predicted by DepthAnything (DA)-v2 and our method to blur the background. As shown in Fig. 12 and in Fig. 4 of the main paper, our method produces sharper fine-grained details (see highlighted regions), which allows part of the plants to be blurred while other parts remain in focus. In contrast, due to over-smoothed depth predictions, DA-v2 fails to separate these structures and thus cannot achieve the same blur effect.

## E. Limitations and Future Work

While our framework demonstrates strong performance for high-resolution depth and normal estimation, it still has several limitations. First, our model assumes mostly diffuse surfaces and struggles with strongly reflective objects such as mirrors, where the mirror region is misinterpreted as a continuation of the surrounding geometry and yields local depth and normal errors (Fig. 9). Second, our current framework is coupled with two specific coarse geometry backbones, namely Depth-Anything V2 for depth and Metric3D V2 for surface normals. As a result, the quality and characteristics of these coarse predictors directly influence the final refined outputs.



(a) Metric3d v2



(b) Ours

Figure 11. **Surface Normal estimation example from ETH3D.**



(a) Depth-Anything v2



(b) Ours

Figure 12. **Application: Lens blur.**

In future work, we aim to relax this dependency and turn our approach into a more generic, plug-and-play refinement module that can be seamlessly attached to a wide range of coarse predictors. We believe such a flexible refinement framework would further broaden the applicability of our method across different tasks, datasets, and deployment scenarios.



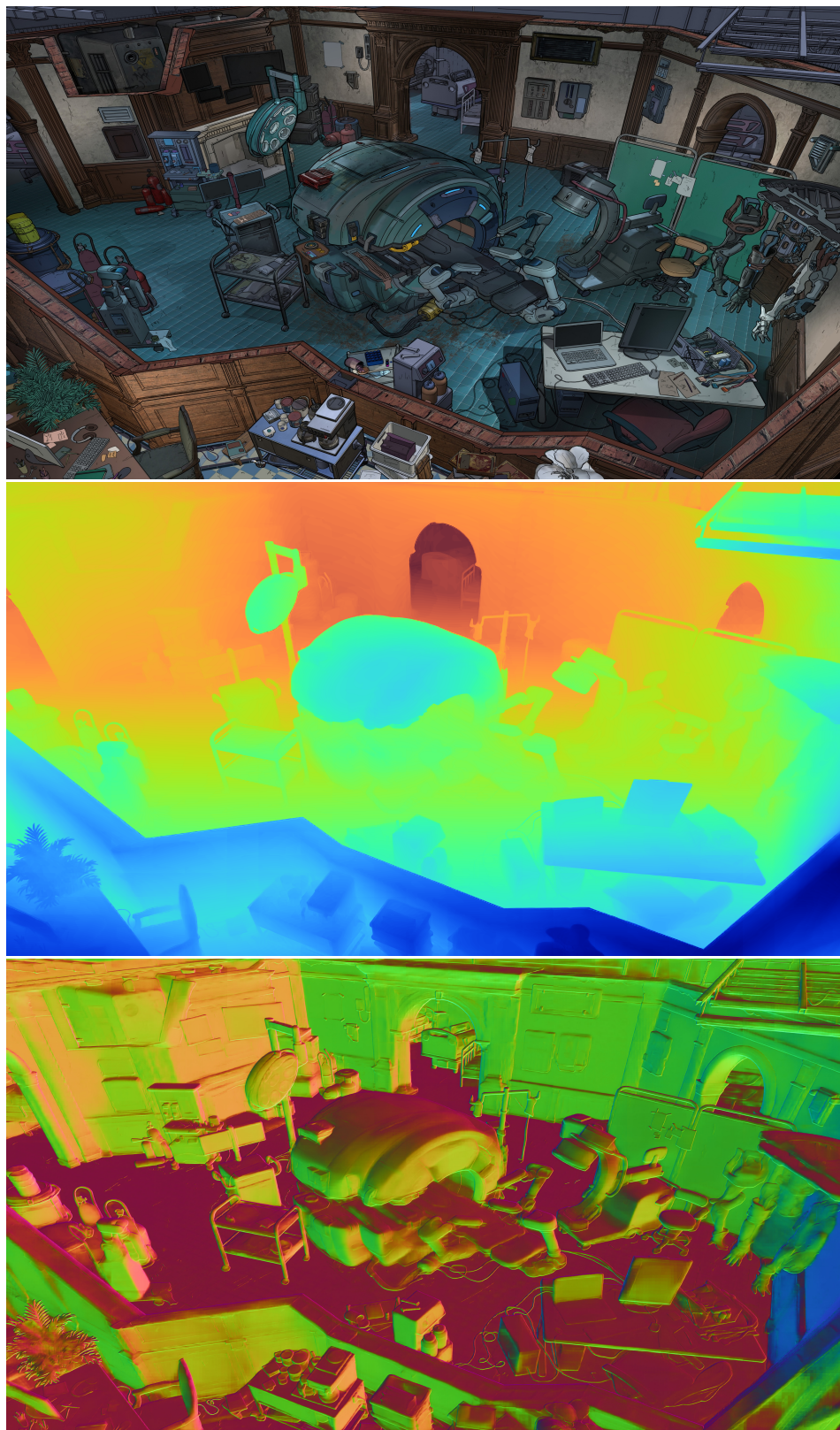


Figure 13. **Out-of-domain 8K manga-style example.** We show an in-the-wild 8K manga-style image, which lies far outside the training distribution, together with our depth and normal predictions. Even under this stylized, out-of-domain setting, our model produces geometrically coherent depth and normals, and the full-resolution visualization allows inspection of fine structures and large-scale consistency.

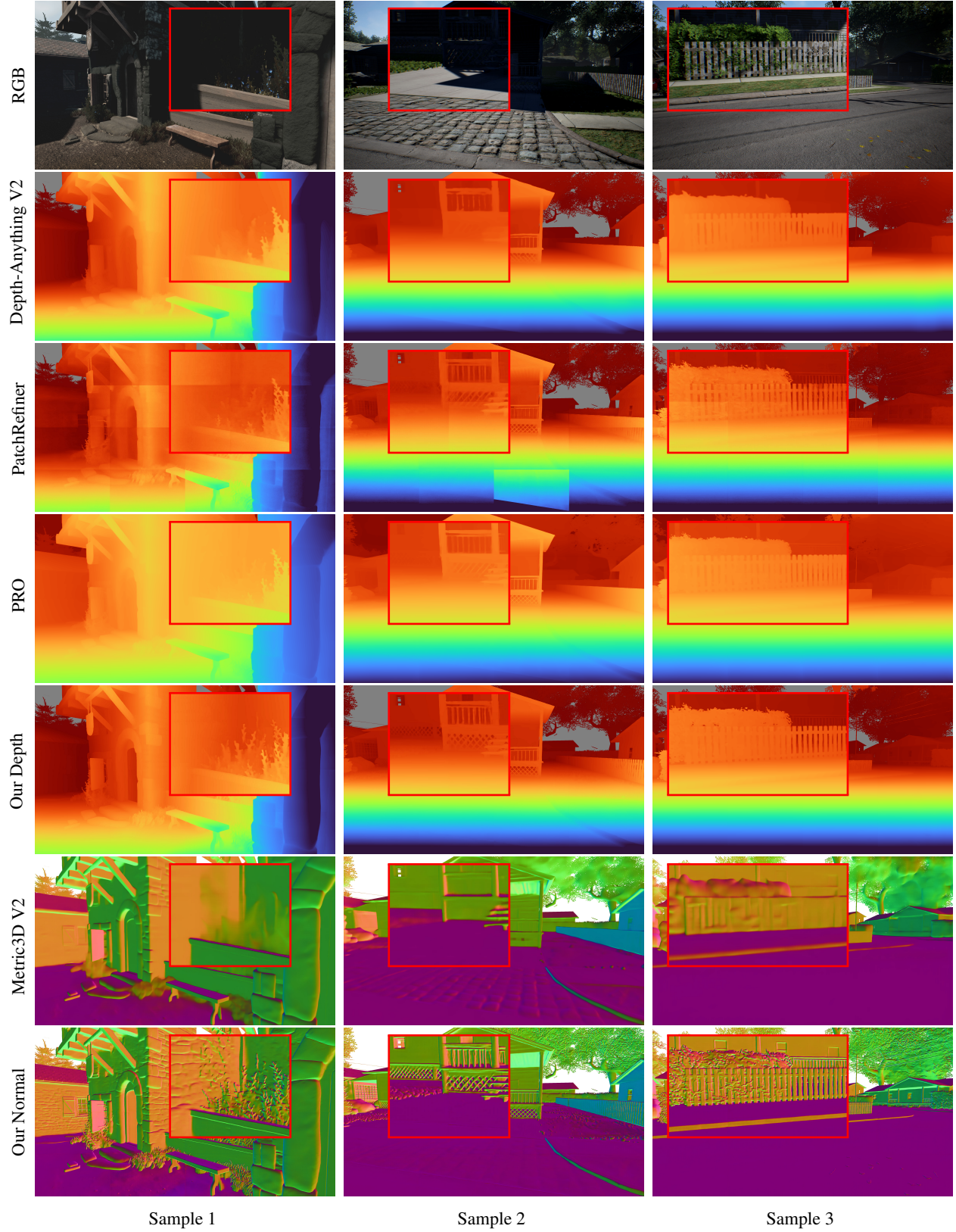


Figure 14. **More qualitative results on UnrealStereo4K.** We show samples from the UnrealStereo4K [24] dataset. Each column corresponds to one scene, and rows show the RGB input, coarse depth prediction by Depth-Anything V2 [29], depth refinements from PatchRefiner [12] and PRO [9], our refined depth prediction, coarse normal prediction by Metric3D V2 [8], and our refined normal prediction. For each column, a consistent zoom-in inset highlights a region of interest across all methods.