Iris: Integrating Language into Diffusion-based Monocular Depth Estimation

Supplementary Material

A. Dataset Details

We train our model on two synthetic datasets, Hypersim [37] and Virtual Kitti [2], and conduct zero-shot evaluations on five additional real-world datasets that were not part of its training data, NYUv2 [41], KITTI [16], ScanNet [6], ETH3D [40], and DIODE [45]. Details of each dataset are provided below.

A.1. Training Datasets

Hypersim [37] is a photorealistic synthetic dataset designed for comprehensive indoor scene understanding, and is introduced since obtaining per-pixel ground truth labels from real images is often challenging or impossible for many essential scene understanding tasks. This dataset is created using a vast collection of synthetic scenes developed by professional artists, resulting in 77,400 images across 461 indoor scenes with detailed per-pixel annotations and corresponding ground truth geometry. HyperSim is built exclusively using publicly accessible 3D assets. It includes complete scene geometry, material properties, and lighting information for each scene. Also, it provides dense per-pixel semantic instance segmentations and comprehensive camera details for each image. Further, it decomposes each image into diffuse reflectance, diffuse illumination, and a non-diffuse residual component that captures view-dependent lighting effects. In terms of training split, for Marigold and E2E-FT, as mentioned in the Experiments Section, we utilize the official dataset split to select approximately 54,000 samples from 365 scenes, and the RGB images and depth maps are resized to a resolution of 480 × 640 pixels. For Lotus, approximately 39,000 samples are selected, and the RGB images and depth maps are resized to a resolution of 576 × 768 pixels. The original distance measurements, defined relative to the focal point, are transformed into standard depth values relative to the focal plane.

Virtual Kitti [2, 13] Virtual KITTI is a photorealistic synthetic video dataset created for training and evaluating computer vision models on various video understanding tasks, including object detection, multi-object tracking, scene-level and instance-level semantic segmentation, optical flow, and depth estimation. The dataset comprises 50 high-resolution monocular videos (a total of 21,260 frames) generated from five distinct virtual urban environments, each presented under varying imaging and weather conditions. These virtual scenes were developed using the Unity game engine and an innovative real-to-virtual cloning technique. The synthetic videos come with precise, fully automatic annotations for 2D and 3D multi-object tracking, as well as per-pixel category, instance, flow, and depth labels. We use its upgraded version, Virtual KITTI 2 [2], which consists of the same five sequence clones as Virtual KITTI, with increased photorealism. It takes advantage of recent advancements in lighting and post-processing within the game engine, making the variations in the virtual sequences more closely mimic real-world changes in conditions. For training, we choose four scenes, comprising around 20,000 samples, and crop the images to match the resolution of the KITTI benchmark [16]. The maximum depth is capped at 80 meters. For all models, the resolution is set to 352 × 1216.

A.2. Evaluation Datasets

NYUv2 [41] dataset comprises 24,231 synchronized RGB images and depth maps at a resolution of 640×480, representing various indoor scenes such as homes, offices, and commercial spaces, captured using a Microsoft Kinect. The standard split includes 249 training scenes and 215 test scenes. For our experiments, we use the official test set. Consistent with prior works [1, 20, 62–64], we exclude samples without valid ground truth, resulting in 654 valid images for evaluation. We perform evaluation on NYUv2 over a depth range spanning from 1×10^{-3} to 10 meters.

KITTI [16, 17] contains 61 driving scenes with research in autonomous driving and computer vision. It contains calibrated RGB images with synchronized point clouds from Velodyne lidar, inertial, GPS information, etc. Following prior works [1, 20, 62–64], we used Eigen split [8]. It consists of 652 testing images after filtering out images without a valid ground truth. We follow the evaluation protocol of [9] for our experiments.

ScanNet [6] is an extensive RGB-D video dataset containing 2.5 million views in more than 1500 scans, annotated with 3D camera poses, surface reconstructions, and instance-level semantic segmentations. Data was collected using an RGB-D capture system (with a Kinect sensor) that includes automated surface reconstruction and crowdsourced semantic annotation. We use the same evaluation configuration of Marigold [20], where 800 images are randomly selected from the 312 official validation scenes for evaluation.

ETH3D [40] is a multi-view stereo and 3D reconstruction benchmark encompassing a diverse range of indoor and outdoor scenes. High-precision laser scanning was used to obtain the ground truth geometry. Images were captured using both a DSLR camera and a synchronized multi-camera rig with varying fields of view. For evaluation, following Marigold [20], we use all 454 samples that include ground

truth depth maps.

DIODE [45] is a dataset containing both indoor and outdoor scenes with high-quality dense depth maps acquired using a FARO Focus S350 laser scanner. The indoor scenes are captured in a variety of settings such as living rooms, bathrooms, and offices, while the outdoor scenes include gardens, plazas, and sidewalks. Following Marigold [20], we use the entire validation split, which encompasses 325 indoor samples and 446 outdoor samples.

B. Implementation Details

B.1. Visualization

When visualizing the ground truth depth map, we apply the same affine transformation we used in training. As described in the Experiments Section, we apply a linear normalization ensuring that the depth values primarily fall within the range $[-1, 1]$. The affine transformation for normalization is defined as:

$$\tilde{y*} = \left(\frac{y^* - y_2}{y_2 - y_{98}} - 0.5 \right) \times 2, \tag{1}$$

where y_2 and y_{98} represent the 2% and 98% percentiles of the depth maps, respectively. Then, we apply min-max normalization to both the ground truth and predicted depth maps, scaling them to integer values within the range [0, 255]. These normalized depth maps are then visualized using the OpenCV MAGMA colormap.

It is important to note that, unlike Marigold [20], which applies linear fitting to the ground truth for error correction, we do not use this approach. While linear fitting can adjust predictions to more closely align with the ground truth, it does not accurately reflect the true distribution of the depth map predictions or provide a clear assessment of prediction quality. Instead, we conduct visualization by applying the same training normalization to the zero-shot evaluation dataset, avoiding linear fitting and its error correction.

B.2. Training Details

Marigold. For Marigold [20], we implemented our method using PyTorch, employing Stable Diffusion v2 [38] as the backbone and maintaining the original pre-training configuration with the v-objective [39]. The training process utilized the DDPM noise scheduler [19] with 1,000 diffusion steps, while at inference time, the DDIM scheduler [43] was employed with 50 sampling steps for faster results. Our training setup spanned 30,000 iterations, with an effective batch size of 32 achieved through gradient accumulation over 16 steps (with a per-step batch size of 2) to fit on a single Nvidia RTX 3090 GPU. We used the Adam optimizer with a learning rate set at $3 \cdot 10^{-5}$ and included random horizontal flipping with a probability of 0.5 as data augmentation. For depth normalization, we employed a scale and shift-invariant method with clipping enabled and set the normalization range between -1.0 and 1.0, using a 0.02 min-max quantile to maintain robustness. This normalization strategy was applied during training and zero-shot evaluations for consistency. The training noise scheduler initialized from the pre-trained Stable Diffusion v2 [38] model maintained a noise strength of 0.9 and incorporated an annealed strategy to progressively reduce noise levels. We saved checkpoints every 50 iterations, with backup, validation, and visualization checkpoints set at intervals of 2,000 iterations. The training process typically converged after approximately 20,000 iterations, though we extended training to 30,000 iterations for thorough coverage. We used mean squared error (MSE) as the loss function, with reduction set to "mean" for averaged loss calculation. A customized iteration-wise exponential scheduler is applied, which adjusts the learning rate iteratively using an exponential decay function. It decays the learning rate to 1% of its initial value over 25,000 iterations with a warmup phase of 100 steps. For text generation, generating a single caption for an image using LLaVA v1.6 on an RTX 3090 takes approximately 3.6 seconds, and we generate 10 captions for each image. For training, we generate 740,000 captions, using 740 GPU hours on an RTX 3090.

Lotus. For Lotus, we implemented our method using PyTorch, employing Stable Diffusion v2 [38] as the backbone and maintaining the original pre-training configuration. The training process utilized the DDPM noise scheduler [19] with 1,000 diffusion steps (inherited from the pre-trained Stable Diffusion v2 configuration), while at inference time, the DDIM scheduler [43] was employed with 1 sampling step. Our training setup spanned 20,000 iterations, with an effective batch size of 36 achieved through gradient accumulation over 3 steps (with a per-step batch size of 4) across 3 Nvidia RTX 3090 GPUs. We used the 8-bit Adam optimizer with a learning rate set at $3{\cdot}10^{-5}$ and a constant learning rate scheduler without warmup. We included random horizontal flipping with a probability of 0.5 as data augmentation. For depth normalization, we employed a truncated disparity method, which was applied during training and zero-shot evaluations for consistency. We saved checkpoints every 500 iterations for Lotus-D and every 1,000 iterations for Lotus-G, with validation checkpoints set at the same intervals. The training process typically converged after approximately 20,000 iterations. The training utilized a mixed dataset strategy, combining Hypersim [37] at a resolution of 576×576 (sampled with 90% probability) and VKITTI [2] at a resolution of 375×375 (sampled with 10% probability). For text generation, we used InternVL3-8B [67] to generate a single caption for each image. The generation process employed a specialized prompt focusing on depth estimation attributes (camera factors, scene properties, relative distances, object types, scales, illuminations,

texture, visual features, occlusions, and boundaries), with a maximum token limit of 77 tokens per caption.

E2E-FT. For E2E-FT [15], we implemented our method using PyTorch, employing Stable Diffusion v2 [38] as the backbone and maintaining the original pre-training configuration with the v-objective [39]. The training process utilized the DDPM noise scheduler [19] with 1,000 diffusion steps (inherited from the pre-trained Stable Diffusion v2 configuration), while at inference time, the DDIM scheduler [43] was employed with 1 sampling step. Our training setup spanned 20,000 iterations, with an effective batch size of 32 achieved through gradient accumulation over 16 steps (with a per-step batch size of 1) across 2 Nvidia RTX 3090 GPUs. We used the Adam optimizer with a learning rate set at $3 \cdot 10^{-5}$ and a customized iteration-wise exponential learning rate scheduler that decays the learning rate to 1% of its initial value over 20,000 iterations with a warmup phase of 100 steps. We included random horizontal flipping with a probability of 0.5 as data augmentation. We employed mixed precision training with bfloat16 (bf16) for improved efficiency. For depth normalization, we employed a quantile-based method with clipping enabled, using the 0.02 and 0.98 quantiles to remove outliers and then normalizing the depth values to the range [-1.0, 1.0]. This normalization strategy was applied during training and zero-shot evaluations for consistency. The training utilized a mixed dataset strategy, combining Hypersim [37] (sampled with 90% probability) and VKITTI [2] (sampled with 10% probability). For text generation, we used InternVL3-8B [67] to generate a single caption for each image. The generation process employed a specialized prompt focusing on depth estimation attributes (camera factors, scene properties, relative distances, object types, scales, illuminations, texture, visual features, occlusions, and boundaries), with a maximum token limit of 77 tokens per caption.

B.3. Evaluation metric

Following the affine-invariant depth evaluation protocol [20, 33, 34, 58, 64], for each image and the predicted relative depth y, we fit a pair of scalars denoting the scale and shift parameters of the transformation: $(\hat{\alpha}, \hat{\beta}) = g_\psi(y, y^*) \in \mathbb{R}^2$. The metric depth prediction is obtained by $\hat{y} = \hat{\alpha} \cdot y + \hat{\beta}$ such that:

$$\psi^* = \arg\min_{\psi} \frac{1}{|M|} \sum_{(i,j)\in\Omega} M(i,j)|\hat{y}(i,j) - y(i,j)| \quad (2)$$

where $\hat{y} = \hat{\alpha} \cdot y + \hat{\beta}$ denotes the predicted metric-scale depth aligned from relative depth y, $(i,j) \in \Omega$ denotes an image coordinate, and $M : \Omega \mapsto \{0,1\}$ denotes a binary mask indicating valid coordinates in the ground truth depth y^* with values greater than zero. Then, we follow [3, 25, 63, 64, 66] to evaluate using first-order threshold accuracy, calculated as:

$$\delta_1 = \%\text{ of } y(i,j) \text{ s.t. } \max\left(\frac{y(i,j)}{y^*(i,j)}, \frac{y^*(i,j)}{y(i,j)}\right) < 1.25 \quad (3)$$

and mean absolute relative error, calculated as:

$$AbsRel = \frac{1}{|M|} \sum_{(i,j)\in\Omega} \frac{|y^*(i,j) - y(i,j)|}{y^*(i,j)} \quad (4)$$

C. Additional Experiments

C.1. Additional visualizations

We provide additional visualization and analysis for indoor scenes in Figure 1 and outdoor scenes in Figure 2. We use several samples in the NYUv2 [41] and KITTI [16, 17] datasets across diverse types of scenes. We have provided examples in captions under each figure. These visualizations demonstrate that leveraging the language enhances the model's ability to understand the geometric characteristics of the specified regions and objects. It shows that language plays a critical role in guiding the model's attention to relevant regions and providing context for improved depth prediction. It highlights subtle or easily overlooked details, such as small objects or instances, and enhances the perception of complex scenes with multiple objects or intricate surfaces. Additionally, language descriptions offer an essential context for partially observed or occluded objects, enabling the model to infer details that visual cues alone might miss. By integrating language, the model achieves a more comprehensive and accurate understanding of scenes, especially in challenging scenarios.

D. Potential Negative Social Impact

Integrating language into depth estimation may be misused in surveillance or privacy-sensitive environments to infer spatial layouts without consent, increasing the risk of unauthorized spatial reconstruction of private spaces. Also, because integrating text relies on natural-language inputs, it inherits biases embedded in textual prompts or language corpora, potentially leading to systematic errors for certain objects or environments and raising fairness concerns for downstream applications such as robotics and AR/VR. Finally, incorrect language descriptions may cause hallucinated geometry in safety-critical domains such as navigation or autonomous systems, where inaccurate depth estimation could lead to hazardous decisions. The high computational cost of training diffusion-based models also contributes to energy consumption, underscoring the importance of responsible development and deployment.

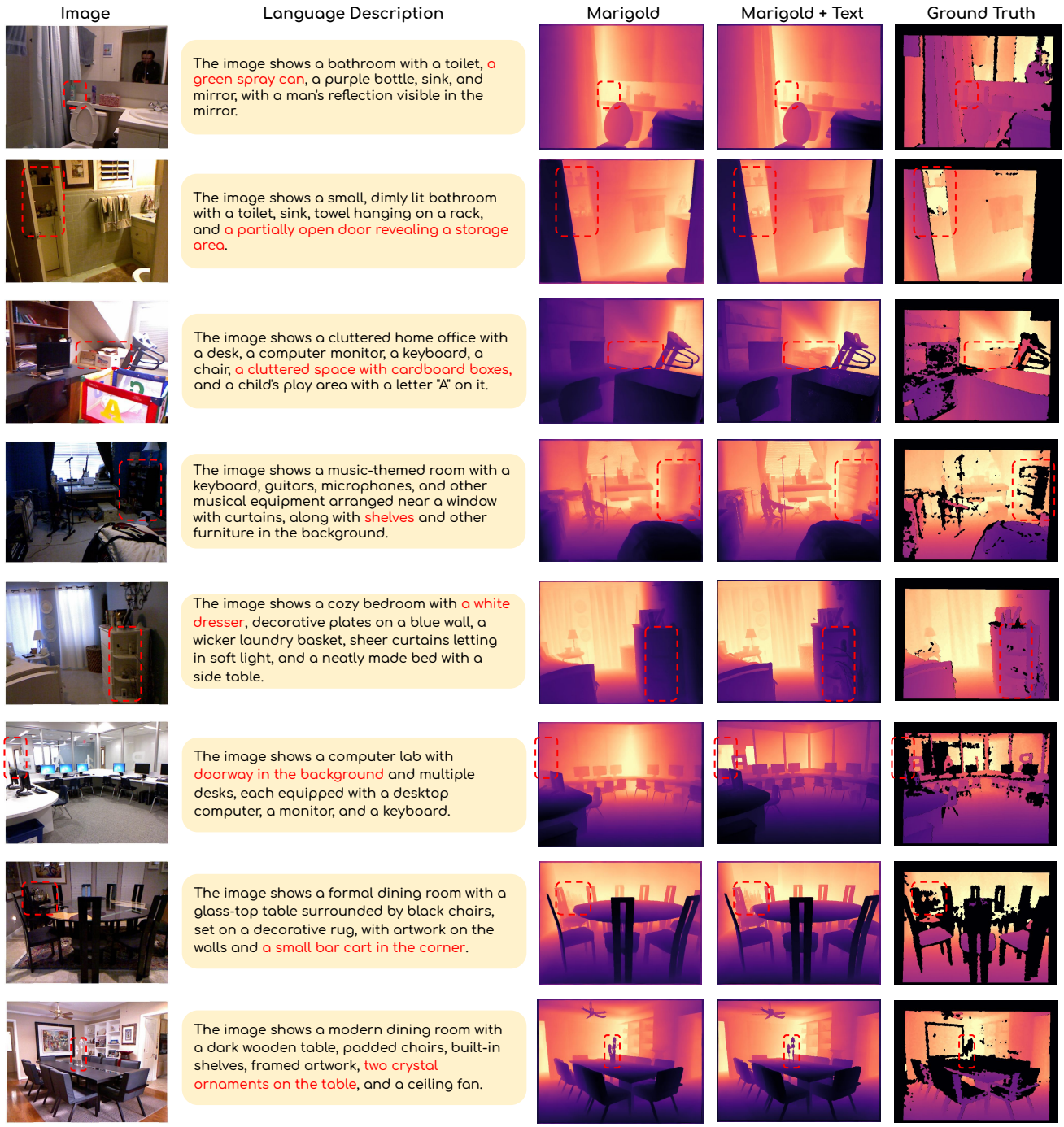

Figure 1. **Additional visualization on NYUv2.** Compared to Marigold, integrating text demonstrates better depth prediction, particularly for instances specified in the language description (highlighted in red text and marked with red boxes). The language description effectively guides the model's attention to relevant regions, especially those easily overlooked by visual cues due to a small size or a transparent texture, such as "a green spray can" in the 1st row and "two crystal ornaments" in the last row. It also improves perception under challenging visual conditions, like "shelves" in the 4th row and "a white dresser" in the 5th row, both of which are under poor illumination and are difficult to tell from visual alone. Additionally, it supports complex reasoning about scene layouts that might be misinterpreted from visual cues alone, such as "a doorway in the background" in the 6th row. Furthermore, it provides critical context for partially observed or occluded objects, such as "a partially open door revealing a storage area" in the 2nd row and "a small bar cart in the corner" in the 7th row.

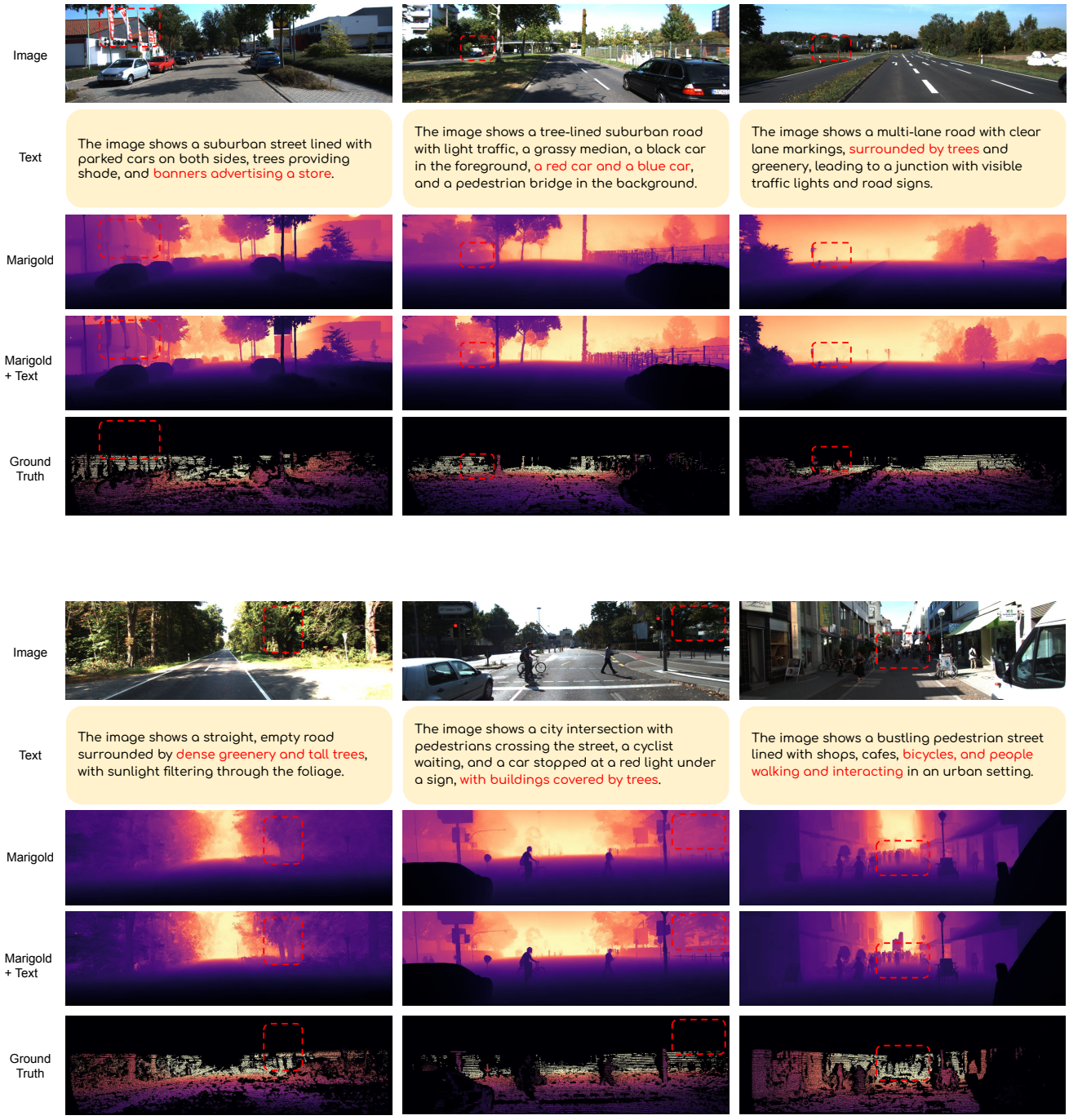

Figure 2. **Additional visualization on KITTI.** Compared to Marigold, integrating text demonstrates superior depth prediction, particularly for instances specified in the language description (highlighted in red text and marked with red boxes). The language description effectively guides the model's attention to relevant regions that might otherwise be overlooked due to their small size or subtle visual cues. Examples include "banners advertising a store" in the upper 1st column and "a red car and a blue car" in the upper 2nd column. Additionally, it enhances perception in complex scenes featuring multiple objects or intricate surfaces. For instance, it accurately captures "surrounded by trees" in the lower 1st column, "dense greenery and tall trees" in the upper 3rd column, and "bicycles, and people walking and interacting" in the lower 3rd column. Furthermore, the language descriptions provide essential context for partially observed or occluded objects, such as "buildings covered by trees" in the lower 2nd column.

C.2. Ablation for prompts to generate text

To study the effect of different prompts for text generation, here we use Marigold [20] and the visual question-answering model LLaVA v1.6 Mistral [26] to generate one text for each training image and testing image. The prompt we use should elicit responses that capture essential details, including the positioning of objects, their interactions, and notable features that influence monocular depth estimation. We generate different prompts using ChatGPT 4o [28], with the prompt:

"Generate prompt for a vision-language model to generate language description for each given image in one sentence. The prompt we use needs to elicit responses that include essential details, such as the positioning of objects, their interactions, and notable features that may impact depth estimation."

We present the results in Table 1, showcasing various language descriptions generated by LLaVA under different prompts. While the performances exhibit some variation, they remain consistently comparable. This demonstrates that diffusion-based depth estimators can be enhanced as long as they are provided with meaningful language descriptions of 3D scenes that resemble natural human descriptions.

C.3. Different denoising steps

As demonstrated in Figure 8 in the main text, we evaluate the Marigold baseline and the Marigold with both training text and inference text, with different denoising steps during inference. The performances are shown in Table 2. With more denoising steps, performance gradually improves. Integrating text consistently outperforms the baseline across various denoising steps, converging in just 10 steps, while the baseline requires 25 steps. This suggests that the language can speed up the denoising process and accelerate convergence.

C.4. Template Prompt Comparison

When training with text, we also consider inference scenarios where user-provided descriptions are unavailable. We therefore explore whether the model can still perform comparably by using either a blank input or standardized template prompts. Specifically, we use the Marigold model trained with text and evaluate the effect of several predefined prompts—ranging from simple ones such as blank string "", simple words like "An image", to more descriptive ones like "A complex 3D scene with varying objects at different distances", as inputs to the diffusion-based depth estimator to maintain its performance.

As presented in Table 3, these template prompts help preserve the model's performance when explicit language input is not feasible. The results show that the model achieves comparable, or even better, performance than the Marigold baseline when using fixed prompts instead of user-provided text. This finding suggests that, even when user-provided descriptions are unavailable, incorporating language during training itself might enhance the depth estimator's generalization and overall performance.

C.5. Ablation on the Number of Text Captions per Image

We test Marigold's performance with different numbers of captions provided for each image, to test whether increasing the number of texts provided during training improves the performance. To generate text descriptions for images, for training images, we use two different versions of LLaVA v1.6 [26], Mistral and Vicuna, each with 5 different prompts, to generate 10 text descriptions for each training image:

- *"Describe the image in one sentence, assuming it's a real-world image."*
- *"Provide a one-sentence description of the image, pay attention to object type, assuming it's a real-world image."*
- *"Capture the essence of the image in a single sentence, pay attention to object relationship, assuming it's a real-world image."*
- *"Condense the image description into one sentence, pay attention to object size, assuming it's a real-world image."*
- *"Express the image in just one sentence, pay attention to the overall layout, assuming it's a real-world image."*

For testing images, we use LLaVA v1.6 Mistral, then prompt this model with:

- *"Describe the image in one sentence, assuming it's a real-world image. Pay close attention to objects, their spatial relationships, and the overall layout."*

This prompt encourages generated responses to include essential details for depth estimation, such as the positioning of objects, their interactions, and notable features that may impact depth estimation. Note that all the training data we use is synthetic data, and by emphasizing "assuming it's a real-world image," we ensure that the descriptions align with the types of inputs and scenarios the model will encounter.

When multiple captions are available for each image, one is randomly selected during training. Shown in Table 4, as the number of captions increases, performance saturates, and the improvement is marginal. One possible explanation is that different captions provide similar descriptions of the scene in terms of attributes essential for depth estimation, such as object types, sizes, spatial relationships, and scene structure, since those captions are prompted to cover all those essential details. Generating additional captions does not introduce additional information, thus leading to only marginal improvements in monocular depth estimation accuracy.

D. Future Work

3D reconstruction, e.g., depth estimation, is an ill-posed inverse problem, where there are insufficient constraints to uniquely infer depth for every pixel, less provide a metric-scale estimate, necessary to support spatial applications. Hence, the introduction of additional information is necessary to resolve such ambiguities. This work considers monocular depth estimation [1, 11, 12, 14, 21, 22, 25, 44, 51, 53, 62, 63, 66] and focuses on evaluating the influence of language on the fidelity of affine-invariant diffusion-based monocular depth estimators. However, we foresee use cases of this work to extend beyond a single image, as insights garnered in our findings are relevant to general 3D reconstruction. Hence, we foresee in relevance in multi-view depth estimation [4, 7, 18, 46–49, 60, 61, 65]. Additionally, as multi-sensory approaches to depth estimation have become common, we also see language being relevant to fusion of camera and lidar [5, 10, 27, 29–32, 35, 52, 54–57, 59] or radar [23, 24, 36, 42, 50]. Furthermore, existing works [63, 64] have demonstrated language to be useful in aligning monocular depth estimators to metric scale. This is beyond the scope of this paper. As promising metric-scale depth estimation has been demonstrated by previous work, albeit with feed-forward networks, we believe that language can likely be used to align diffusion-based models; we leave this for future work.

References

[1] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4009–4018, 2021. 1, 7

[2] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. arXiv preprint arXiv:2001.10773, 2020. 1, 2, 3

[3] Wenjie Chang, Yueyi Zhang, and Zhiwei Xiong. Transformer-based monocular depth estimation with attention supervision. In 32nd British Machine Vision Conference (BMVC 2021), 2021. 3

[4] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1538–1547, 2019. 7

[5] Younjoon Chung, Hyoungseob Park, Patrick Rim, Xiaoran Zhang, Jihe He, Ziyao Zeng, Safa Cicek, Byung-Woo Hong, James S. Duncan, and Alex Wong. Eta: Energy-based test-time adaptation for depth completion. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025. 7

[6] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5828–5839, 2017. 1

[7] Ruxiao Duan and Alex Wong. Evidential neural radiance fields. In Proceedings of the Computer Vision and Pattern Recognition Conference, 2026. 7

[8] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE international conference on computer vision, pages 2650–2658, 2015. 1

[9] 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. 1

[10] Vadim Ezhov, Hyoungseob Park, Zhaoyang Zhang, Rishi Upadhyay, Howard Zhang, Chethan Chinder Chandrappa, Achuta Kadambi, Yunhao Ba, Julie Dorsey, and Alex Wong. All-day depth completion. In 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2024. 7

[11] Xiaohan Fei, Alex Wong, and Stefano Soatto. Geo-supervised visual depth prediction. IEEE Robotics and Automation Letters, 4(2):1661–1668, 2019. 7

[12] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2002–2011, 2018. 7

[13] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 4340–4349, 2016. 1

[14] Suchisrit Gangopadhyay, Jung-Hee Kim, Xien Chen, Patrick Rim, Hyoungseob Park, and Alex Wong. Extending foundational monocular depth estimators to fisheye cameras with calibration tokens. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025. 7

[15] Gonzalo Martin Garcia, Karim Abou Zeid, Christian Schmidt, Daan de Geus, Alexander Hermans, and Bastian Leibe. Fine-tuning image-conditional diffusion models is easier than you think. arXiv preprint arXiv:2409.11355, 2024. 3

[16] 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, pages 3354–3361. IEEE, 2012. 1, 3

[17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 32(11):1231–1237, 2013. 1, 3

[18] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2495–2504, 2020. 7

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020. 2, 3

[20] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. arXiv preprint arXiv:2312.02145, 2023. 1, 2, 3, 6

[21] Dong Lao, Yangchao Wu, Tian Yu Liu, Alex Wong, and Stefano Soatto. Sub-token vit embedding via stochastic resonance transformers. In International Conference on Machine Learning. PMLR, 2024. 7

[22] Dong Lao, Fengyu Yang, Daniel Wang, Hyoungseob Park, Samuel Lu, Alex Wong, and Stefano Soatto. On the viability of monocular depth pre-training for semantic segmentation. In European Conference on Computer Vision. Springer, 2024. 7

[23] Huadong Li, Minhao Jing, Wang Jin, Shichao Dong, Jiajun Liang, Haoqiang Fan, and Renhe Ji. Sparse beats dense: Rethinking supervision in radar-camera depth completion. In European Conference on Computer Vision, pages 127–143. Springer, 2024. 7

[24] Han Li, Yukai Ma, Yaqing Gu, Kewei Hu, Yong Liu, and Xingxing Zuo. Radarcam-depth: Radar-camera fusion for depth estimation with learned metric scale. In 2024 IEEE International Conference on Robotics and Automation (ICRA), pages 10665–10672. IEEE, 2024. 7

[25] Ce Liu, Suryansh Kumar, Shuhang Gu, Radu Timofte, and Luc Van Gool. Va-depthnet: A variational approach to single image depth prediction. arXiv preprint arXiv:2302.06556, 2023. 3, 7

[26] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 6

[27] Tian Yu Liu, Parth Agrawal, Allison Chen, Byung-Woo Hong, and Alex Wong. Monitored distillation for positive congruent depth completion. In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II, pages 35–53. Springer, 2022. 7

[28] OpenAI. Chatgpt-4: Conversational ai model, 2024. Accessed via OpenAI's platform for generating and refining content. 6

[29] Hyoungseob Park, Anjali Gupta, and Alex Wong. Test-time adaptation for depth completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 20519–20529, 2024. 7

[30] Hyoungseob Park, Runjian Chen, Patrick Rim, Dong Lao, and Alex Wong. Orcas: Unsupervised depth completion via occluded region completion as supervision. The Fourteenth International Conference on Learning Representations, 2026.

[31] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16, pages 120–136. Springer, 2020.

[32] Jin-Hwi Park, Chanhwi Jeong, Junoh Lee, and Hae-Gon Jeon. Depth prompting for sensor-agnostic depth estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9859–9869, 2024. 7

[33] 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. 3

[34] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In Proceedings of the IEEE/CVF international conference on computer vision, pages 12179–12188, 2021. 3

[35] Patrick Rim, Hyoungseob Park, Ziyao Zeng, Younjoon Chung, and Alex Wong. Protodepth: Unsupervised continual depth completion with prototypes. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 6304–6316, 2025. 7

[36] Patrick Rim, Hyoungseob Park, Vadim Ezhov, Jeffrey Moon, and Alex Wong. Radar-guided polynomial fitting for metric depth estimation. In Proceedings of the Computer Vision and Pattern Recognition Conference, 2026. 7

[37] 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 Proceedings of the IEEE/CVF international conference on computer vision, pages 10912–10922, 2021. 1, 2, 3

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10684–10695, 2022. 2, 3

[39] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512, 2022. 2, 3

[40] 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, pages 3260–3269, 2017. 1

[41] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12, pages 746–760. Springer, 2012. 1, 3

[42] Akash Deep Singh, Yunhao Ba, Ankur Sarker, Howard Zhang, Achuta Kadambi, Stefano Soatto, Mani Srivastava, and Alex Wong. Depth estimation from camera image and mmwave radar point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9275–9285, 2023. 7

[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502, 2020. 2, 3

[44] Rishi Upadhyay, Howard Zhang, Yunhao Ba, Ethan Yang, Blake Gella, Sicheng Jiang, Alex Wong, and Achuta Kadambi. Enhancing diffusion models with 3d perspective geometry constraints. ACM Transactions on Graphics (TOG), 42(6):1–15, 2023. 7

[45] 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. 1, 2

[46] Daniel Wang, Patrick Rim, Tian Tian, Dong Lao, Alex Wong, and Ganesh Sundaramoorthi. Ode-gs: Latent odes for dynamic scene extrapolation with 3d gaussian splatting. The Fourteenth International Conference on Learning Representations, 2026. 7

[47] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14194–14203, 2021.

[48] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 5294–5306, 2025.

[49] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 20697–20709, 2024. 7

[50] Yiran Wang, Jiaqi Li, Chaoyi Hong, Ruibo Li, Liusheng Sun, Xiao Song, Zhe Wang, Zhiguo Cao, and Guosheng Lin. Tacodepth: Towards efficient radar-camera depth estimation with one-stage fusion. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 10523–10533, 2025. 7

[51] Alex Wong and Stefano Soatto. Bilateral cyclic constraint and adaptive regularization for unsupervised monocular depth prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5644–5653, 2019. 7

[52] Alex Wong and Stefano Soatto. Unsupervised depth completion with calibrated backprojection layers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 12747–12756, 2021. 7

[53] Alex Wong, Safa Cicek, and Stefano Soatto. Targeted adversarial perturbations for monocular depth prediction. Advances in neural information processing systems, 33: 8486–8497, 2020. 7

[54] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. IEEE Robotics and Automation Letters, 5(2): 1899–1906, 2020. 7

[55] Alex Wong, Safa Cicek, and Stefano Soatto. Learning topology from synthetic data for unsupervised depth completion. IEEE Robotics and Automation Letters, 6(2):1495–1502, 2021.

[56] Alex Wong, Xiaohan Fei, Byung-Woo Hong, and Stefano Soatto. An adaptive framework for learning unsupervised depth completion. IEEE Robotics and Automation Letters, 6 (2):3120–3127, 2021.

[57] Yangchao Wu, Tian Yu Liu, Hyoungseob Park, Stefano Soatto, Dong Lao, and Alex Wong. Augundo: Scaling up augmentations for monocular depth completion and estimation. In European Conference on Computer Vision, pages 274–293. Springer, 2024. 7

[58] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10371–10381, 2024. 3

[59] Yanchao Yang, Alex Wong, and Stefano Soatto. Dense depth posterior (ddp) from single image and sparse range. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3353–3362, 2019. 7

[60] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In Proceedings of the European Conference on Computer Vision (ECCV), pages 767–783, 2018. 7

[61] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5525–5534, 2019. 7

[62] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3916–3925, 2022. 1, 7

[63] Ziyao Zeng, Daniel Wang, Fengyu Yang, Hyoungseob Park, Stefano Soatto, Dong Lao, and Alex Wong. Wordepth: Variational language prior for monocular depth estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9708–9719, 2024. 3, 7

[64] Ziyao Zeng, Yangchao Wu, Hyoungseob Park, Daniel Wang, Fengyu Yang, Stefano Soatto, Dong Lao, Byung-Woo Hong, and Alex Wong. Rsa: Resolving scale ambiguities in monocular depth estimators through language descriptions. Advances in neural information processing systems, 37, 2024. 1, 3, 7

[65] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. arXiv preprint arXiv:2410.03825, 2024. 7

[66] Renrui Zhang, Ziyao Zeng, Ziyu Guo, and Yafeng Li. Can language understand depth? In Proceedings of the 30th ACM International Conference on Multimedia, pages 6868–6874, 2022. 3, 7

[67] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. arXiv preprint arXiv:2504.10479, 2025. 2, 3

Method	NYUv2		KITTI		ETH3D	
	$\delta_1\uparrow$	AbsRel↓	$\delta_1\uparrow$	AbsRel↓	$\delta_1\uparrow$	AbsRel↓
Marigold	95.7	6.1	89.7	10.4	95.4	6.9
"An image"	95.7	6.1	89.8	10.7	95.1	6.8
Template A	95.8	6.0	90.3	10.6	95.5	6.4
Template B	95.7	6.1	90.5	10.7	95.6	6.5
Template C	**95.9**	6.0	90.2	10.6	95.3	6.6
Template D	95.6	**5.8**	90.4	10.5	95.4	6.4
Template E	95.8	5.9	90.3	10.7	95.6	6.7
Template F	**95.9**	6.0	90.5	10.5	95.4	6.6
Template G	95.8	6.1	90.3	10.6	95.5	**6.3**
Template H	95.7	5.9	90.2	10.7	**95.7**	**6.3**
Template I	95.8	6.1	**90.6**	10.5	95.4	6.6
Template J	95.8	6.0	90.5	10.6	95.6	6.4
Template K	95.7	**5.8**	90.4	10.5	95.5	**6.3**
Ours	**95.9**	5.9	**90.6**	**10.4**	**95.7**	6.5

Template A: "Describe the image in one sentence. Explain the image by identifying key objects and their distances from the viewpoint, noting any perspective lines or depth cues that indicate the three-dimensional structure of the scene."
Template B: "Describe the image in one sentence. Describe the image by specifying the foreground, midground, and background elements, with emphasis on their relative depth, distances, and spatial relationships within the scene."
Template C: "Describe the image in one sentence. Describe the image by detailing the foreground, midground, and background objects, emphasizing their relative distances and spatial positioning within the scene."
Template D: "Describe the image in one sentence. Provide an in-depth description of the image, focusing on the scale and depth of each visible object and how they overlap or are spaced from one another."
Template E: "Describe the image in one sentence. Analyze the image by discussing the size and arrangement of objects, their positions relative to one another, and any changes in texture or clarity that indicate varying depths across the scene."
Template F: "Describe the image in one sentence. Describe the scene with attention to depth, specifying which elements appear closer or farther from the viewer and how shadows or lighting contribute to the perception of depth."
Template G: "Describe the image in one sentence. Highlight the depth relationships in the image by describing which objects are in the foreground, which are in the background, and how their relative sizes help convey distance."
Template H: "Describe the image in one sentence. Focus on any natural or man-made structures in the image and describe how their orientation and placement give a sense of depth or perspective."
Template I: "Describe the image in one sentence. Describe how elements like roads, pathways, or fences create leading lines that guide the viewer's eye into the depth of the scene."
Template J: "Describe the image in one sentence. Explain how differences in lighting or shadowing in the image indicate which parts are nearer or further away from the observer."
Template K: "Describe the image in one sentence. Analyze the spatial arrangement of the main objects and describe any overlapping or occlusion that suggests depth relationships between them. "
Ours: "Describe the image in one sentence, assuming it's a real-world image, pay more attention to objects, their spatial relationships, and the overall layout."

Table 1. **Ablation for prompts to generate language description.** Prompts are used to prompt LLaVA to generate language descriptions for each image. While the performances among different prompts may vary, they remain consistently comparable, as long as they are meaningful and mimic human descriptions. Marigold in the first row is trained without text.

Steps	NYUv2		KITTI		ETH3D		ScanNet	
	$\delta_1\uparrow$	AbsRel↓	$\delta_1\uparrow$	AbsRel↓	$\delta_1\uparrow$	AbsRel↓	$\delta_1\uparrow$	AbsRel↓
Marigold								
1	48.8	33.8	25.2	59.1	50.1	37.5	60.3	25.7
2	78.7	16.9	72.2	18.0	86.0	12.7	72.1	19.2
3	92.6	8.2	77.7	15.3	92.0	8.6	87.1	11.4
4	94.2	7.0	80.4	14.1	93.6	7.6	90.7	9.3
5	94.7	6.7	82.4	13.4	94.3	7.3	91.9	8.6
10	95.1	6.4	86.5	11.9	94.9	6.9	93.2	7.8
15	95.1	6.4	87.4	11.6	95.0	6.9	93.4	7.7
20	95.2	6.3	88.0	11.4	95.0	6.9	93.7	7.5
25	95.3	6.3	88.2	11.3	95.0	6.9	93.7	7.5
50	95.3	6.3	88.5	11.2	95.0	6.9	93.8	7.5
Marigold + Text (Training & Inference)								
1	48.8	33.8	25.2	59.1	50.1	37.5	60.3	25.7
2	83.2	14.1	74.7	16.9	86.5	12.0	75.0	17.8
3	94.3	7.1	81.2	13.9	93.0	7.7	90.6	9.5
4	95.5	6.3	84.7	12.4	94.7	6.9	93.5	7.8
5	95.7	6.0	87.0	11.6	95.3	6.6	91.9	8.6
10	96.0	5.9	89.9	10.5	95.7	6.5	94.9	6.8
15	96.0	5.9	90.1	10.4	95.8	6.5	94.9	6.8
20	96.0	5.9	90.3	10.4	95.8	6.5	94.9	6.7
25	96.0	5.9	90.3	10.4	95.8	6.5	94.9	6.7
50	96.0	5.9	90.3	10.4	95.8	6.5	94.9	6.7

Table 2. **Performance for different denoising steps.** Integrating text consistently outperforms the baseline across various denoising steps, with a significantly faster convergence speed for the diffusion process.

Method	NYUv2		KITTI		ETH3D	
	$\delta_1\uparrow$	AbsRel↓	$\delta_1\uparrow$	AbsRel↓	$\delta_1\uparrow$	AbsRel↓
Blank text input	95.5	6.2	89.3	10.9	95.0	6.9
"An image"	95.7	6.1	89.8	10.7	95.1	6.8
Template A	**95.8**	**6.0**	**89.9**	10.7	95.3	6.8
Template B	95.7	6.1	89.8	10.7	95.2	6.8
Template C	**95.8**	6.1	89.8	10.7	**95.3**	**6.8**
Marigold*	95.7	6.1	89.7	10.7	**95.4**	6.9
With text input	95.9	5.9	90.6	10.4	95.7	6.5

Template A: 'A complex 3D scene with varying objects at different distances."
Template B: 'A structured environment with intricate patterns and designs that create depth and guide the eye through various focal points."
Template C: 'An elaborate scene with overlapping objects that create a sense of distance and spatial hierarchy within the environment."

Table 3. **Inference using fixed template text input.** The results show that the model achieves comparable, or even better, performance than the Marigold baseline when using fixed prompts instead of user-provided text. This finding suggests that, even when user-provided descriptions are unavailable, incorporating language during training itself might enhance the depth estimator's generalization and overall performance.

Method	NYUv2		KITTI		ETH3D	
	$\delta_1\uparrow$	AbsRel↓	$\delta_1\uparrow$	AbsRel↓	$\delta_1\uparrow$	AbsRel↓
1 Caption Per Image	95.9	5.9	90.6	10.4	95.7	6.5
2 Captions Per Image	95.9	5.9	90.5	10.4	95.7	6.5
5 Captions Per Image	96.0	5.9	90.4	10.4	95.8	6.5
10 Captions Per Image	96.0	5.9	90.3	10.4	95.8	6.5

Table 4. **Training with different numbers of text captions per image.** For images annotated with multiple captions, a caption is randomly sampled for each training iteration. While adding more captions initially slightly improves performance, the benefit quickly saturates, yielding only minor gains beyond a certain point.